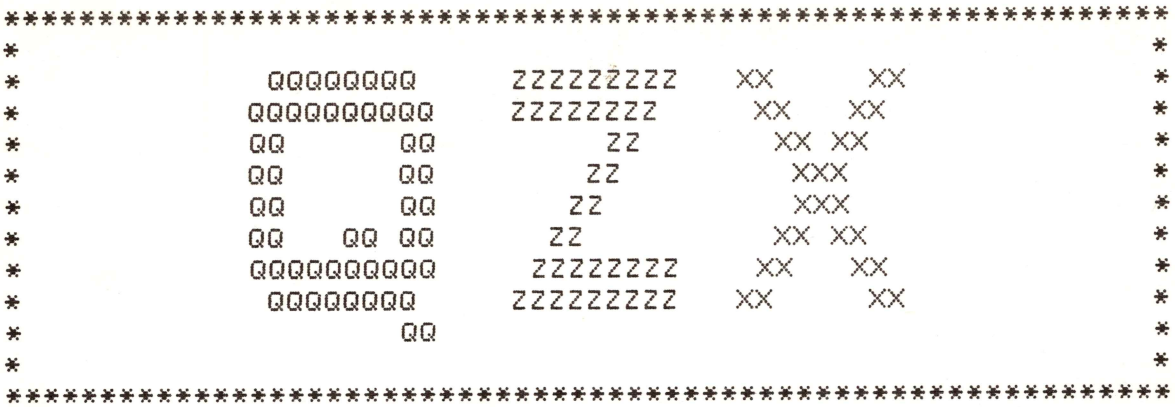


The Journal Covering Amateur Radio & Sinclair Computers
ZX-80, Micro Ace, ZX-81, & Timex Sinclair 1000/1500
2025 O'Donnell Drive
Las Cruces, New Mexico
88001

Alex F. Burr,K5XY
Publisher

Ambrose 'Bo' Barry,W4GHV/5
Editor

IN THIS ISSUE

Volume 1 Number 12 ($2.50/issue) Nov 1983

* ZX81- Baudot TTY Machine Interface & Program, WA0MRG
* Complete ZX81 (& Spectrum) RTTY System, OZ1BII
* RTTY Terminal Unit, G4IDE
* ZX81 RTTY Send Program, W9HLQ
* The Basic RTTY Station, K5XY
* RTTY Literature, K5XY
* RTTY Frequencies, K5XY
* ASCII/Baudot/Sinclair Code Conversions, K5XY
* Editor's Doings, W4GHV
* Unclassified Ads
* Products in Review
  AF8J RTTY interface/program
  NU4V RTTY interface/program
* QZX Friends
* Bits & Bytes

Stand-by for December issue, its in the works

ZX 81 BAUDOT INTERFACE - WAØMRG

THOSE WHO WOULD LIKE TO INTERFACE THEIR ZX 81 TO THAT OLD TELETYPE MACHINE SITTING IN THE GARAGE ARE FACED WITH SEVERAL PROBLEMS.

FIRST IS THE PROBLEM OF HARDWARE. SECONDLY, THE CHARACTER SETS DO NOT MATCH. ONCE BOTH OF THESE PROBLEMS ARE SOLVED, THE BAUDOT TTY MACHINE MAKES A FINE ADDITION TO THE COMPUTER.

FOR HARDWARE THERE ARE A NUMBER OF SOLUTIONS: USE THE CASSETE PORT TO DRIVE THE TTY LOOP (EASILY DONE WITH SOFTWARE BUT REQUIRES A DEMODULATOR); BUY OR BUILD A FULL RS-232 INTERFACE AND THEN MATE IT TO THE TTY LOOP (COSTLY) OR YOU MIGHT TRY THE METHOD GIVEN HERE.

FOR LOW COST AND SIMPLICITY THIS SYSTEM WILL BE HARD TO BEAT. THE HARDWARE REQUIRES ONLY 2 CHEAP CHIPS AND 7 CONNECTIONS TO THE ZX 81. AS WRITTEN THE PROGRAM WILL ALLOW THE TTY MACHINE TO BE USED TO PRINT PROGRAM LISTINGS ALTHOUGH SIMPLE MODIFICATIONS WILL TURN IT INTO A RTTY SEND OR WORD PROCESSING PROGRAM.

THE PROBLEM OF CHARACTER SET IS SOLVED WITH SOFTWARE AND A LITTLE IMAGINATION. FOR INSTANCE, THE EQUAL SIGN IS PRINTED AS A COLON, THE PLUS SIGN IS PRINTED AS THE "HASH" OR POUND SIGN AND THE GREATER THAN AND LESS THAN SYMBOLS ALONG WITH SEVERAL OTHERS ARE SPELLED OUT.

PROGRAM INSTRUCTIONS

SINCE THIS PROGRAM CONTAINS A LARGE AMOUNT OF MACHINE LANGUAGE, THE USUAL PRECAUTIONS HOLD. MAKE FREQUENT SAVES OF THE LISTING AS YOU GO. STEPS 100 THROUGH 1090 ARE FOR LOADING THE MACHINE CODE AND SHOULD BE DELETED AFTER THE CODE IS ENTERED.

YOU MAY WISH TO CHANGE THE NAME OF THE PROGRAM. IF SO MAKE SURE THE NEW NAME HAS 9 CHARACTERS (INCLUDING SPACES) INSIDE THE QUOTES IN 1 REM. FOLLOW 2 REM WITH 43 PERIODS, 3 REM WITH 64 PERIODS AND 4 REM WITH 30 PERIODS. THEN ENTER STEPS UP TO 1090.

AT THIS POINT RUN THE PROGRAM AND ENTER THE MACHINE CODE AS DIRECTED. (SEE CHARTS FOR LISTING.)

YOUR RESULTS WILL LOOK SIMILIAR TO THE PROGRAM LISTING EXCEPT THE COMPUTER WILL SHOW GRAPHIC SYMBOLS IN SOME PLACES. AFTER THE MACHINE CODE IS ENTERED AND CHECKED STEPS 100 THROUGH 1090 ARE DELETED. NOW ENTER THE REMAINING PROGRAM.

MAKING IT ALL WORK

IF YOUR TTY LOOP DOES NOT USE AN OPTOCOUPLER IT WOULD BE WISE TO USE ONE. (DO NOT CONNECT THE 74HC00 DIRECTLY TO THE LOOP.)

THE PROGRAM IS SET UP FOR 100 WPM BAUDOT WITH A 2 BIT LENGTH STOP PULSE. TO CHANGE TO 60 WPM THEN POKE 16564,18.

IF THE TTY MACHINE RUNS OPEN USING THE Q OUTPUT OF THE 74HC00 THEN USE THE "NOT Q".

WHEN WRITING A PROGRAM, PUT IT BETWEEN STEPS 60 AND 9599. TO KEEP THIS ROUTINE FROM PRINTING ITSELF ALONG WITH THE DESIRED PROGRAM, CHANGE THE VALUE 9900 IN STEP 9638 TO 9598. THEN ONLY STEPS BEFORE 9599 WILL BE PRINTED.

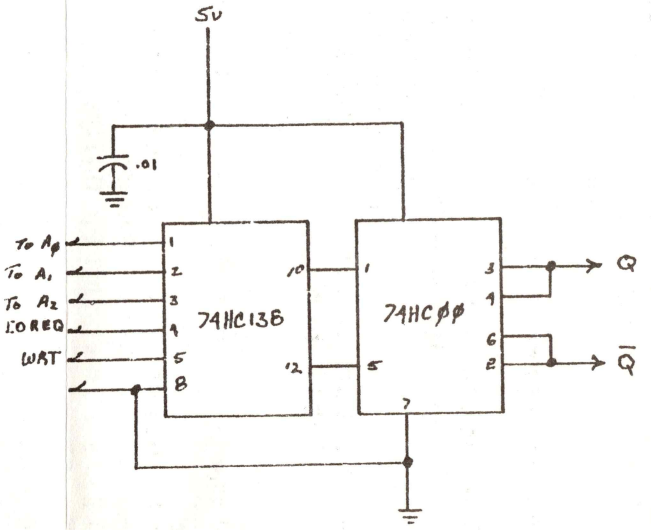
BECAUSE THERE ARE MINOR DIFFERENCES IN THE CHARACTER SETS ON BAUDOT MACHINES, SOME CHANGES IN THE LOOKUP TABLE MAY BE NECESSARY. FOR EXAMPLE, ON SOME MACHINES A SHIFTED J IS THE BELL WHILE ON OTHER MACHINES IT IS AN APOSTROPHE. THESE CHANGES WILL HAVE TO BE MADE THE PARTICULAR MACHINE.

DISADVANTAGES

FIRST, OF COURSE, THERE ARE NO GRAPHICS. THESE ARE PRINTED AS "?". ALSO IT TAKES SOME GETTING USED TO THE SYMBOLS FOR EQUALS AND THE LIKE. AND THE PROGRAM IS SLOW. WHILE THE BAUD RATE IS 74.2 (100 WPM) THE ACTUAL CHARACTER RATE IS MUCH SLOWER.

IN SPITE OF THESE SHORT COMINGS, THE PROGRAM AND INTERFACE WORK QUITE WELL. WITH SIMPLE MODIFICATIONS TO THE SOFTWARE, A RTTY SEND OR WORD PROCESSING PROGRAM CAN BE DEVELOPED. (ONE SUCH PROGRAM WAS USED TO PREPARE THIS.

DALLAS WILLIAMS
WAØMRG/Ø
P.O. BOX 1
SEDGWICK, COLO.
80749



TTL LEVEL SERIAL BAUDOT TO TTY LOOP VIA OPTO coupl. (see Text)

RTTY INTERFA. for ZX-81

EDITOR'S NOTE: THE HEADINGS FOR THE FIRST FOUR ARTICLES CAN BE FOUND A MERE 19 PAGES FROM HERE! (SORRY)

```
  1 REM "WORD PROS"
  2 REM :??PEEK ?LN RNDACS TK?P
EEK ?/?PEEK ??4 LOAD PEEK ?LN RND
LN RNDTAN -"2 PRINT !4 CLEAR #4 S
AVE TAN
  3 REM ?          L?DITJM3?IO?X
YGWQRN?EKPBASTID?HUOAFJMWGSQRE?KB
YNXPL
  4 REM E?RNDLN ??E?RND7??7??6
?RND?77TAN E POUND RND7??TAN

 10 PRINT "DO YOU WISH PRINT OF
 PROGRAM? (Y OR N)"
 20 INPUT A$
 30 IF A$:"N" THEN GOTO 60
 40 IF A$ LESS THAN  GREATER TH
AN "Y" THEN GOTO 20
 50 GOTO 9600
 60 CLS
 100 REM DELETE STEPS 100 THRU 1
090 WHEN MACHINE CODE HAS BEEN EN
TERED
 110 PRINT "MACHINE CODE ENTRY R
OUTINE"
 120 PRINT "BAUDOT SEND"
 130 FOR N:16531 TO 16573
 140 GOSUB 1000
 190 NEXT N
 200 CLS
 210 PRINT "LOOKUP TABLE ENTRY R
OUTINE"
 220 FOR N:16580 TO 16643
 230 GOSUB 1000
 240 NEXT N
 250 CLS
 260 PRINT "CONVERSION ROUTINE"
 270 FOR N:16650 TO 16679
 280 GOSUB 1000
 290 NEXT N
 300 STOP
1000 REM
1020 PRINT AT 4,0;N
1050 PRINT "ENTER VALUE TO BE PO
KED"
1060 INPUT P
1070 PRINT AT 5,0;"
      "
1080 POKE N,P
1090 RETURN
9599 STOP
9600 FAST
9601 GOSUB 9760
9602 GOSUB 9760
9603 REM X EQUALS TIMES
9604 REM XX EQUALS POWER FUNCTIO
N
9606 REM : EQUALS EQUAL SIGN
9608 REM POUND SIMBOL EQUALS PLU
S
9610 LET A$:""
9612 LET E:0
9614 LET B:63
9616 POKE 16507,125
9618 POKE 16508,64
9619 GOSUB 9621
9620 GOTO 9700
9621 CLS
9622 LET V:USR 16650
9624 LET A$:""
9626 LET Q:USR 16673
9628 FOR I:0 TO Q#791-(33 X PEEK
 16442)

9630 IF PEEK I LESS THAN  GREATE
R THAN 118 THEN LET A$:A$#CHR$ PE
EK I
9632 NEXT I
9634 GOSUB 9760
9636 CLS
9638 IF V LESS THAN 9900 THEN RE
TURN
9640 GOSUB 9760
9642 GOSUB 9760
9644 GOSUB 9760
9646 GOSUB 9760
9648 STOP
9700 IF E GREATER THAN 32 THEN G
OSUB 9760
9702 IF CODE A$:192 THEN LET A$:
CHR$ 11#A$(2 TO )
9704 IF CODE A$ GREATER THAN 63
THEN LET A$:""#A$(2 TO )

9706 IF CODE A$ LESS THAN 38 THE
N LET B:63
9708 IF CODE A$ LESS THAN 38 OR
CODE A$ GREATER THAN 63 THEN GOTO
 9714
9710 IF B:63 THEN GOTO 9746
9712 GOTO 9720
9714 IF CODE A$:0 OR B:59 THEN G
OTO 9746
9716 LET B:59
9718 GOTO 9722
9720 LET B:63
9722 POKE 16532,B
9724 LET U:USR 16531
9726 IF CODE A$:19 THEN LET A$:"
 LESS THAN "#A$(2 TO )
9728 IF CODE A$:18 THEN LET A$:"
 GREATER THAN "#A$(2 TO )
9736 IF CODE A$:23 THEN LET A$:"
 X "#A$(2 TO )
9738 IF CODE A$ LESS THAN 192 AN
D CODE A$ GREATER THAN 119 THEN L
ET A$:"?"#A$(2 TO )
9740 IF CODE A$:216 THEN LET A$:
" XX "#A$(2 TO )
9742 IF CODE A$ GREATER THAN 0 A
ND CODE A$ LESS THAN 11 THEN LET
A$:"?"#A$(2 TO )
9744 IF CODE A$:12 THEN LET A$:"
 POUND "#A$(2 TO )
```

9746 POKE 16532,PEEK(16580 #CODE A$)

```
9748 LET U:USR 16531
9750 LET A$:A$(2 TO )
9752 IF A$:"" THEN GOSUB 9621
9754 LET E:E#1
9756 GOTO 9700
9760 LET C$:" GREATER THAN  LESS
 THAN  POUND "
9762 POKE 16532,PEEK (16580#CODE
 C$)
9764 LET U:USR 16531
9766 LET C$:C$(2 TO )
9768 IF C$ LESS THAN  GREATER TH
AN "" THEN GOTO 9762
9770 LET E:0
9772 RETURN
```

BAUDOT SEND MACHINE LAUGUAGE (2 REM)

MEMORY	POKE
16531	14
16532	2
16533	6
16534	6
16535	211
16536	3
16537	205
16538	179
16539	64
16540	175
16541	203
16542	57
16543	48
16544	4
16545	211
16546	5
16547	24
16548	2
16549	211
16550	3
16551	5
16552	32
16553	239
16554	211
16555	5
16556	205
16557	179
16558	64
16559	205
16560	179
16561	64
16562	201
16563	22
16564	11
16565	30
16566	245
16567	29
16568	32
16569	253
16570	21
16571	32
16572	248
16573	201

(3 REM)

MEMORY	POKE
16580	132
16581	2
16582	0
16583	0
16584	0
16585	0
16586	0
16587	0
16588	0
16589	0
16590	0
16591	49
16592	2
16593	41
16594	46
16595	57
16596	47
16597	50
16598	31
16599	8
16600	46
16601	52
16602	131
16603	5
16604	61
16605	62
16606	44
16607	60
16608	54
16609	55
16610	51
16611	1
16612	42
16613	48
16614	53
16615	39
16616	38
16617	56
16618	131
16619	57
16620	46
16621	41
16622	1
16623	45
16624	58
16625	52
16626	38
16627	43
16628	47
16629	50
16630	60
16631	44
16632	56
16633	54
16634	55
16635	42
16636	5
16637	48
16638	39
16639	62
16640	51
16641	61
16642	53
16643	49

CONVERSION ROUTINE (4 REM)

MEMORY	POKE
16650	42
16651	123
16652	64
16653	205
16654	69
16655	7
16656	42
16657	123
16658	64
16659	35
16660	35
16661	78
16662	35
16663	70
16664	35
16665	9
16666	34
16667	123
16668	64
16669	70
16670	35
16671	78
16672	201
16673	42
16674	12
16675	64
16676	35
16677	68
16678	77
16679	201

A Complete rtty transceive system for the ZX81 AND Spectrum
by Henning Andresen OZ1BII.

INTERFACE Fig.1. Theory of operation -- please read introduction further on in this section

The masterchip is an INTEL 8251A USART (Universal Syncron Asyncron Receiver Transmitter). This chip makes all the hard work with Parallel-seriel/ Syncronisation/baudrate interfacing and does it without any trouble.
The chip has two CONTROL and one TX-BUFFER register for writing and it has one STATUS and one RX-BUFFER register for reading.
When using the USART, you have to start with a master reset. Then you must give it a control-word imédiately after the reset. This CONTROL-word determines the controlling of the chip: SYNCRON/ASYNCRON, BAUD RATE FACTOR, CHARACTER LENGHT, PARITY ENABLED/DISABLED, PARITY EVEN/ODD and NUMBER OF STOPBITS FOR TX.
When the first control-word are in the USART, the chip is in a stable mode and you can now chose which MODE you want by sending another control-word to it. This second word gives you the ability to chose various modes:
TX ENABLED, $\overline{\text{DTR}}$ LOW, RX ENABLED, CONTINOUSLY SPACE(BREAK), INTERNAL ERROR RESET, $\overline{\text{RTS}}$ LOW, MASTER RESET (for CONTROL-input), HUNT MODE.
Hunt Mode is for syncron operation only.
Whenever the computer wants to use the usart, it will have to know what it is doing at the moment. You cant send a character to the USART when the TX buffer allready are full and you can't get a character that ain't yet in the RX-buffer. Maybee there are an overrun error, so you miss a character without notice or get a half character. Don't give up ! The USART has the great overview and takes care of it all. For this purpose the STATUS register can be read any time you want to. It will give you the following informations:
TX READY FOR DATA, RX READY FOR DATA, TX EMPTY (SENT LAST CHARACTER IN CHIP), PARITY ERROR (RX), OVERRUN ERROR (RX), FRAMING ERROR (MISSING STOPBIT, RX), BREAK DETECTOR (CONTINOUSLY SPACE, RX), DATA SET READY ($\overline{\text{DSR}}$=0).
Now you can make it run. Here is a litle routine to initialize it to RX-MODE and Baudot RTTY:

```
out CONTROL,0       3 times zero to be shure it is in
out CONTROL,0       the MODE-state.
out CONTROL,0
out CONTROL,40h     Hex 40 is a RESET to CONTROL-State.
out CONTROL,82h     Sets chip to BAUDOT (Async/5 bit/45.45/1½ Stop).
NOP                 Wait (Not used in BASIC)
out CONTROL,14h     Now in MODE-State and setting it in RX-Enable and
                    make a INTERNAL ERROR RESET.
```


After this you can use a routine to read the STATUS register to see when the USART has a character for you to read. This can be done like:

```
in A,CONTROL    Get the STATUS byte.
bit 1,a         Is bit one Set ?
jr Z,NOCHR      No, then jump to something else (No character).
in a,DATA       Yes, get the character from data-adress.
```

If you want to transmit characters you should change the last number in the initialization routine to read: out CONTROL,1
The STATUS read would be the same except that it is bit zero you are interested in and that the last line should be: out DATA,character.

The other chip on fig 1 is the BAUD RATE generator made of a 555'er.
The frequence are measured on pin 3 and should be as follows:
727.2 Hz for 45.45 baud, 800 Hz for 50 baud and 1760 for 110 baud operation.

MODEM

It is very simple but it will make the job.
This I will make short.
First you got the XR2206 which is an AFSK generator with superb data. The 2 tones are adjusted until you get the right frequencies. The Resistor on pin 7 are MARK (High tone) and pin 8 are space (LOW tone). If you use SSB on the transmitter the only adjustments are to get the shift right (170 Hz), but if you want to receive on other frequencies than the standard tones, you must adjust them to the same as them you want to use for receiving (this because you are going to use the generator when adjusting the demodulator). The only really important point is to get the shift exactly 170 Hz !
The second chip is a DEMODULATOR with Phase-locked loop system especially designed for data communications. The XR2211 will except analog signals from 3V to as low as 2 mV so there should be a good chance to pick up weak signals.
Adjustment:
Connect the output from XR2206 to the analog input. Connect a signal-generator with a 50 Hz/4 volt signal to the input of the AFSK-generator (2206). If you don't have a signal-generator use the computer/interface sending continously F's. Then connect a scope to the output of the demodulator and adjust the resistor at pin 12 until you have a symmetrical picture on the scope. If you don't have a scope you can use a voltmeter (DC) at the output and adjust until it shows about half the voltage of the power supply (4.5V).

0Z1BII rtty ZX81 program.

```
    1 REM BBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
    2 REM
    3 REM RTTY FOR ZX-81 !!!!!!!!
    4 REM THIS ARE THE CODES FOR
    5 REM LINE 1.USE A HEX-LOADER
    6 REM 00 3A 82 40 D3 3F C9 3A
 82 40 D3 1F C9 0E 3F 06 00 ED 4
8 C9 0E 1F 06 00 ED 48 C9 AF D3
3F 00 D3 3F 00 D3 3F 3E 40 D3 3F
 3E 82 D3 3F 3E 34 D3 3F C9 3E 0
1 D3 3F C9
    8 REM          GL Vy 73 0Z1BII
    9 REM ======================
   10 LET 0=0
   20 LET B$="INPUT B$ NOW!"
   30 RANDOMIZE USR 16541
  100 REM [illegible]
  105 CLS
  110 PRINT "
  120 PRINT "                [illegible]
  130 PRINT "                [illegible]
  135 PRINT "                [illegible]
  140 PRINT "
  145 IF 0=0 THEN GO TO 5000
  200 PRINT AT 9,0;"1: RX-ROUTINE
"
  210 PRINT AT 11,0;"2: TX-ROUTIN
E"
  220 PRINT AT 13,0;"      A$=ONE
LINE RYRY"
  230 PRINT AT 15,0;"      B$= ";B$
  240 IF 0=10 THEN INPUT B$
  243 LET 0=5
  245 PRINT AT 15,7;B$
  250 PRINT AT 19,0;"WICH ROUTINE
 (1 OR 2)?"
  300 IF INKEY$="1" THEN GO SUB 1
000
  310 IF INKEY$="2" THEN GO SUB 2
000
  315 IF 0=7 THEN GO TO 100
  320 GO TO 300
 1010 CLS : PRINT "
[illegible]
 1020 POKE 16514,52
 1030 RANDOMIZE USR 16515
 1040 LET A=0
 1050 LET IN=USR 16527
 1060 LET C=0
 1100 IF IN=135 THEN LET C=USR 16
534
 1110 IF IN>135 THEN GO TO 1050
 1115 IF INKEY$=" " THEN GO TO 60
00
 1150 IF C<>27 THEN GO TO 1160
 1155 LET A=100
 1156 GO TO 1050
 1160 IF C<>31 THEN GO TO 1170
 1165 LET A=0
 1166 GO TO 1050
 1170 GO SUB 1400+A+C*2
 1200 GO TO 1050
 1400 RETURN
 1402 PRINT "E";
 1403 RETURN
 1404 PRINT CHR$ 118;
 1405 RETURN
 1406 PRINT "A";
 1407 RETURN
 1408 PRINT " ";
 1409 RETURN
 1410 PRINT "S";
 1411 RETURN
 1412 PRINT "I";
 1413 RETURN
 1414 PRINT "U";
 1416 RETURN
 1418 PRINT "D";
 1419 RETURN
 1420 PRINT "R";
 1421 RETURN
 1422 PRINT "J";
 1423 RETURN
 1424 PRINT "N";
 1425 RETURN
 1426 PRINT "F";
 1427 RETURN
 1428 PRINT "C";
 1429 RETURN
```

0Z1BII ZX81 rtty comtinued.

```
 1430 PRINT "K";
 1431 RETURN
 1432 PRINT "T";
 1433 RETURN
 1434 PRINT "Z";
 1435 RETURN
 1436 PRINT "L";
 1437 RETURN
 1438 PRINT "W";
 1439 RETURN
 1440 PRINT "H";
 1441 RETURN
 1442 PRINT "Y";
 1443 RETURN
 1444 PRINT "P";
 1445 RETURN
 1446 PRINT "Q";
 1447 RETURN
 1448 PRINT "O";
 1449 RETURN
 1450 PRINT "B";
 1451 RETURN
 1452 PRINT "G";
 1454 RETURN
 1456 PRINT "M";
 1457 RETURN
 1458 PRINT "X";
 1459 RETURN
 1460 PRINT "V";
 1461 RETURN
 1462 RETURN
 1500 RETURN
 1502 PRINT "3";
 1503 RETURN
 1504 PRINT CHR$ 118;
 1505 RETURN
 1506 PRINT "-";
 1507 RETURN
 1508 PRINT " ";
 1509 RETURN
 1510 PRINT "'";
 1511 RETURN
 1512 PRINT "8";
 1513 RETURN
 1514 PRINT "7";
 1516 RETURN
 1518 PRINT "*";
 1519 RETURN
 1520 PRINT "4";
 1521 RETURN
 1522 PRINT "$";
 1523 RETURN
 1524 PRINT ",";
 1525 RETURN
 1526 PRINT "AA";
 1527 RETURN
 1528 PRINT ":";
 1529 RETURN
 1530 PRINT "(";
 1531 RETURN
 1532 PRINT "5";
 1533 RETURN
 1534 PRINT "+";
 1535 RETURN
 1536 PRINT ")";
 1537 RETURN
 1538 PRINT "2";
 1539 RETURN
 1540 PRINT "OE";
 1541 RETURN
 1542 PRINT "6";
 1543 RETURN
 1544 PRINT "0";
 1545 RETURN
 1546 PRINT "1";
 1547 RETURN
 1548 PRINT "9";
 1549 RETURN
 1550 PRINT "?";
 1551 RETURN
 1552 PRINT "AE";
 1554 RETURN
 1556 PRINT ".";
 1557 RETURN
 1558 PRINT "/";
 1559 RETURN
 1560 PRINI "=";
 1562 RETURN
```

```
 2020 CLS
 2025 PRINT "     [illegible]                TX-ROU
TINE
 2030 PRINT AT 3,0;"                NOW SE
NDING A$+B$"
 2040 POKE 16514,1
 2050 RANDOMIZE USR 16515
 2100 LET AB=0
 2110 LET AA=0
 2120 LET A$=CHR$ 13+CHR$ 118+"RY
RYRYRYRYRYRYRYRYRYRYRYRYRYRYRY
RYRYRYRYRYRYRYRYRYRYRYRYRYRYRY"
 2130 LET A$=A$+CHR$ 118+CHR$ 13+
B$+CHR$ 13
 2140 FOR N=1 TO LEN A$
 2150 LET C=CODE A$(N)
 2160 GO SUB 2300
 2170 PAUSE 5
 2203 IF INKEY$=" " THEN GO TO 60
00
 2204 NEXT N
 2205 CLS
 2210 PRINT "     [illegible]          REPEAT F/ROUT
INE SETTING
 2250 IF INKEY$<>"" THEN GO TO 22
50
 2255 IF INKEY$="" THEN GO TO 225
5
 2260 LET C=CODE INKEY$
 2270 IF C=64 THEN RETURN
 2280 GO SUB 2300
 2290 GO TO 2250
 2300 PRINT CHR$ C;
 2302 IF C=118 THEN GO TO 2700
 2309 IF C<38 THEN LET AA=100
 2310 IF C>37 THEN LET AA=0
 2320 IF AB<AA THEN GO SUB 5000
 2330 IF AB>AA THEN GO SUB 5100
 2340 LET AB=AA
 2360 GO SUB 2400+C*2
 2370 POKE 16514,P
 2375 GO SUB 2800
 2380 RANDOMIZE USR 16521
 2399 RETURN
 2400 LET P=4          2464 LET P=10
 2401 RETURN           2465 RETURN
 2420 LET P=0          2466 LET P=16
 2421 RETURN           2467 RETURN
 2422 LET P=5          2468 LET P=21
 2423 RETURN           2469 RETURN
 2426 LET P=31         2470 LET P=7
 2427 RETURN           2471 RETURN
 2428 LET P=14         2472 LET P=6
 2429 RETURN           2473 RETURN
 2430 LET P=25         2474 LET P=24
 2431 RETURN           2475 RETURN
 2432 LET P=15         2476 LET P=3
 2433 RETURN           2477 RETURN
 2434 LET P=18         2478 LET P=25
 2435 RETURN           2479 RETURN
 2436 LET P=0          2480 LET P=14
 2437 RETURN           2481 RETURN
 2438 LET P=30         2482 LET P=9
 2439 RETURN           2483 RETURN
 2440 LET P=17         2484 LET P=1
 2441 RETURN           2485 RETURN
 2442 LET P=3          2486 LET P=13
 2443 RETURN           2487 RETURN
 2444 LET P=11         2488 LET P=26
 2445 RETURN           2489 RETURN
 2446 LET P=29         2490 LET P=20
 2447 RETURN           2491 RETURN
 2448 LET P=0          2492 LET P=6
 2449 RETURN           2493 RETURN
 2450 LET P=12         2494 LET P=11
 2451 RETURN           2495 RETURN
 2452 LET P=28         2496 LET P=15
 2453 RETURN           2497 RETURN
 2454 LET P=22         2498 LET P=18
 2455 RETURN           2499 RETURN
 2456 LET P=0          2500 LET P=28
 2457 RETURN           2501 RETURN
 2458 LET P=23         2502 LET P=12
 2459 RETURN           2503 RETURN
 2460 LET P=19         2504 LET P=24
 2461 RETURN           2505 RETURN
 2462 LET P=1          2506 LET P=22
 2463 RETURN           2507 RETURN
```



```
 2505 LET P=23
 2509 RETURN
 2510 LET P=10
 2511 RETURN
 2512 LET P=5
 2513 RETURN
 2514 LET P=16
 2515 RETURN
 2516 LET P=7
 2517 RETURN
 2518 LET P=30
 2519 RETURN
 2520 LET P=19
 2521 RETURN
 2522 LET P=29
 2523 RETURN
 2524 LET P=21
 2525 RETURN
 2526 LET P=17
 2527 RETURN
 2700 GO SUB 2800
 2710 POKE 16514,6
 2720 RANDOMIZE USR 16521
 2730 GO SUB 2800
 2740 POKE 16514,8
 2750 RANDOMIZE USR 16521
 2760 GO SUB 2800
 2770 POKE 16514,2
 2780 RANDOMIZE USR 16521
 2790 RETURN
 2800 LET CN=USR 16527
 2810 IF CN=133 THEN RETURN
 2820 GO TO 2800
```



```
 5000 PRINT
 5005 PRINT "THIS PROGRAM WILL  S
END  AND   RECEIVE BAUDOT RTTY
CODES WITH  YOUR ZX 81 COMPUTER.
"
 5007 PRINT
 5010 PRINT "AT RX-MODE (1) THE P
RINTING WILLBE SCROLLING UP THE
DISPLAY"
 5020 PRINT "PRESSING ""SPACE"" O
NLY WILL  BRING YOU BACK TO THE
MENU."
 5025 PRINT
 5030 PRINT "AT TX-MODE (2) IT ST
ARTS WITH   SENDING ONE LINE RYR
YS AND THEN THE MESSAGE IN B$ YO
U MADE."
 5040 PRINT "B$ MUST BE MADE FIRS
T TIME YOU  ARE IN MENU-MODE. "
 5050 PRINT "IN TX-MODE PRES ""RN
D"" FOR RETURNTO MENU."
 5060 PRINT AT 20,0;"PUSH ANY KEY
!"
 5070 IF INKEY$="" THEN GO TO 507
0
 5080 LET 0=10
 5090 GO TO 100
 8000 LET 0=7
 8010 RETURN
```

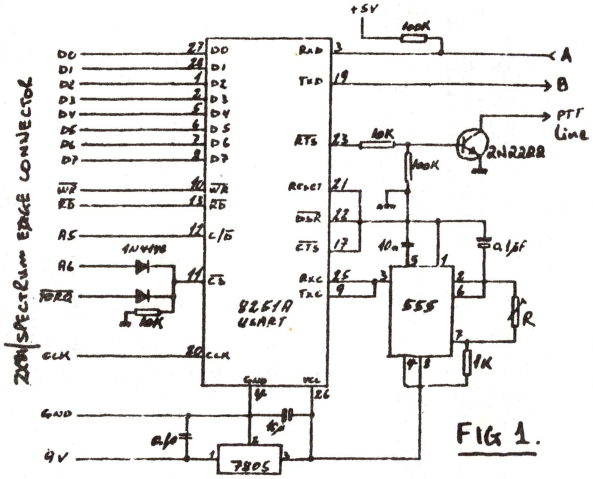


FIG 1.

OZ1B11 rtty

R= 20K multiturn trimpot.
R= 9.4K = 45.5 baud
R= 8.5K = 50 baud
R= 3.5K = 110 baud.

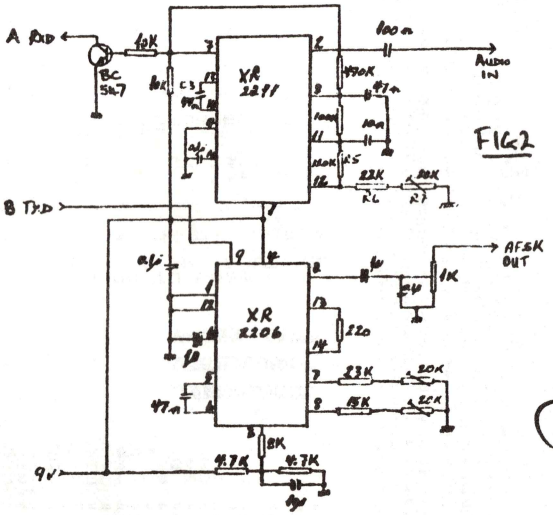


FIG 2

with given values the centre-frequency is about 750Hz.

This ckt is copyrighted
© OZ1B11 1983

Timex/Sinclair HAMS : HAWG WILD Software(TM) announces a new series of programs for Radio Amateurs (This is a direct result of our on-going author search). First programs in the HAWG WILD "Ham-Hacker"(TM) series are as follows :

*MORSE CODE(16K)/TWO K CODE
Code practice with built-in audio tone
by Gary Fox KC9MR....$14.95

*MINIMUF 3.5-Maximum Usable Freq. radio propagation program A MUST for DXers
by Sandy Blaize W5TVW.$17.95

*CE AMP -Common Emitter Amp
Computer Aided Des. from SYNCWARE Co(TM),El Monte, CA by Fred Nachbaur...$19.95

Does your program belong here?
10% off till Dec. 15,'83
add $1.00 each P&H.AR. res.
add 5% tax.

HAWG WILD Software

SOFTWARE SYSTEMS COMPONENTS HARDWARE

P.O. Box 7668
Little Rock, Arkansas 72217

THE G4IDE RTTY TERMINAL UNIT

This terminal unit was, of course, designed to be used with my ZX81 RTTY program. However it is equally suitable for use with any micro-computer with a TTL compatible I/O port.
On receive the output to the port from the TU is a TTL low level signal ('0') when no signal or mark is received, and a TTL high level signal ('1') when a space is received. On transmit it sends mark when it is receiving '1' from the port and space when it is receiving '0'. Should you ever wish to use the TU with a system using the opposite logic then a TTL inverter such as a 74LS04 can be put on the receive output and pin 13 on IC2 on the AFSK board connected to pin 3 on IC1 instead of to pin 6 as shown.

The Receive Board

This board is quite conventional in design. IC1 with ZD1 and ZD2 forms a limiting amplifier. IC2 and IC3 are the mark and space filters and IC4 acts as a comparator on the filter outputs. ZD3 clips the output to a TTL compatible level. S1 allows the board to handle 2 different frequency shifts. IC5 is a buffer amp to drive the tuning meter. S2 inverts the filters for receiving reverse shift signals.

The AFSK Board

This is a totally novel design. Having built 3 published designs and found none of them satisfactory, I designed my own. IC3 is the oscillator; I use a 567 because although not intended for this purpose it provides a very stable output. Note I am only using the free running oscillator, not the phase-locked loop. The frequency of oscillation is determined by the value of the resistance between pins 5 and 6 and the capacitance between pin 6 and ground. For space the resistor comprises R2 and VR2. For mark VR1 is switched in parallel with VR2 by the CMOS switch IC2. IC4 is a simple low pass filter/buffer amp on the output. IC1 is a buffer on the input and enables the logic to be easily changed as explained above. A regulator is included on this board so it can be powered from the +12V supply from the PSU board.

The PSU Board

This is very straight-forward. It has ample capacity to power any other 'goodies' you may want to add to the TU in the future. All the components are over-rated for the current drawn, but I believe in power supplies that run cold.

Construction Notes

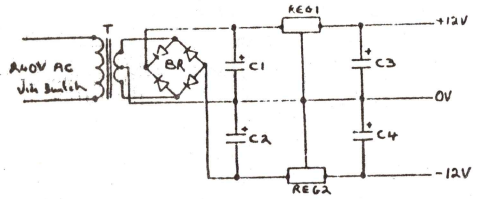
The following symbols are used on the Veroboard drawings:-
Track side: X = cut the track. o = soldered connection.
Component side: O = Veropin. •——• = wire link.
Building the TU really amounts to following the pictures! The Veroboard layouts can undoubtedly be improved on. My technique tends to be to plant the components as I build the circuit and only to draw the boards when I have finished. If you are good at making PCB's then this would give a neater unit.

All the components are quite standard and can be obtained from most dealers. Personally I always use Watford Electronics, however they do not seem to stock the multi-turn trimpots which I got from Maplin. The holes for some of the polyester capacitors are quite close together and I used capacitors shaped rather than . For the receive board I have included a diagram with nothing on it except the various Veropin connections. C14 is fixed directly across the back of the meter. Note that the receive board has 2 linked tracks down the middle for 0V.
The whole unit should be enclosed in a metal case to avoid RF feedback. Do not forget that a terminal unit is an audio unit and it is possible to create 'earth loops' just as with hi-fi.
When you have finished building, check the power supply voltages before you connect it to the other boards! NB - the 7812 and 7912 are mounted opposite ways round on the board. Check the resistance across the power lines on the AFSK and receive boards to show up any potentially disastrous shorts before you connect the boards to the PSU. ⊛

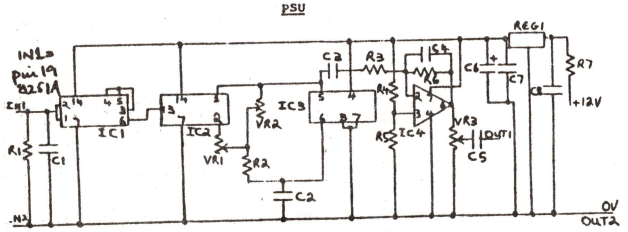
⊛ Note - on the receive board, VR1 to 4 each have one pin through a cut in the track - ie. cannot be soldered.

Alignment

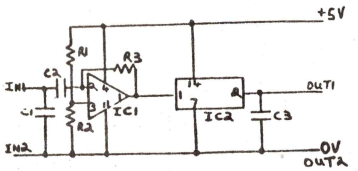
This is simple if you first construct the port and the small board for the frequency counter, as described later.
Power up the AFSK board, turn VR3 up to maximum and check with an earpiece that there is output. Using the ZX81 as a frequency counter, adjust VR2 for a frequency of 1275Hz. Now make a temporary link from IN1 on the AFSK board to +5V and adjust VR1 for a frequency of 1445Hz. Recheck the first frequency, but it should not have changed.
To align the receive board, temporarily set the AFSK board to give an output of 1445Hz and connect its output to the receive input. Adjust VR1 or VR2 (depending on which position of S1 you are going to use for narrow shift) for a peak on the tuning meter. Use VR4 to set the meter to give a reading of about 90% full scale. Now set the AFSK board to 1275Hz and adjust VR3 for a peak on the meter; this should be virtually the same as the previous peak. If there is a large difference between the two peaks and you are sure you have not made a mistake on the board, then you will have to fit a 100k preset on the input to the filter giving the highest peak, as shown in the diagrams. The frequency to which you set the other preset depends on what shift you wish to have available as an alternative.
To check the output from the board connect the input to the external speaker socket on your rig. Put a voltmeter on pin 6 of IC4 and tune slowly backwards and forwards across a steady carrier. The voltage should swing between +12V and -12V. Finally check that the voltage at OUT1 swings between +4.7V and 0V.



PSU



AFSK Board



*SEE issue 8

Frequency Counter Board *

COMPONENTS

All resistors are ¼watt/5%. All capacitors, unless stated otherwise, are polyester. Veroboard sizes are not critical and can be obtained from the drawings.

PSU Board

T: 15-0-15V, 100mA transformer	BR: 1A/50V bridge rectifier
C1: 1000µF/25V electrolytic	C2: 1000µF/25V electrolytic
C3: 100µF/25V electrolytic	C4: 100µF/25V electrolytic
REG1: 7812 voltage regulator	REG2: 7912 voltage regulator

AFSK Board

R1: 4k7	R5: 10k
R2: 8k2	R6: 100k
R3: 10k	R7: 100R
R4: 10k	
C1: 10nF ceramic	C5: 0.22µF
C2: 47nF	C6: 100µF/25V
C3: 10nF	C7: 10nF ceramic
C4: 0.1µF	C8: 10nF ceramic
IC1: 74LS00	IC3: NE567
IC2: 4066	IC4: 741
VR1: 10k multi-turn trimpot	VR2: 5k multi-turn trimpot
VR3: 10k miniature horizontal preset	REG1: 5V/100mA voltage regulator

Receive Board

R1: 10k	R10: 10k
R2: 1k	R11: 100k
R3: 1M	R12: 100k
R4: 4k7	R13: 100k
R5: 150k	R14: 100k
R6: 150k	R15: 1k
R7: 180k	R16: 220k
R8: 180k	R17: 4k7
R9: 10k	R18: 100k
C1: 0.22µF	C8: 10nF
C2: 22nF	C9: 10nF
C3: 22nF	C10: 0.47µF tantalum
C4: 22nF	C11: 0.47µF tantalum
C5: 22nF	C12: 10nF
C6: 22nF	C13: 10nF
C7: 22nF	C14: 220µF/25V

VR1, VR2, VR3: 250R miniature horizontal preset
D1, D2, D3, D4, D5: 1N914
ZD1, ZD2: 5V1/400mW Zener diode
ZD3: 4V7/400mW Zener diode
IC1, IC2, IC3, IC4, IC5: 741 op amp
S1: 1 pole/2 way miniature switch
S2: 2 pole/2 way miniature switch
M: 1mA meter

VR4: 5k miniature horizontal preset

Frequency Counter Board

R1: 10k	R2: 10k
R3: 1M	
C1: 10nF ceramic	C2: 0.1µF
C3: 10nF ceramic	
IC1: LM324	IC2: 74LS14

SYSTEM INTERCONNECTIONS

If you use this TU with my port design and my R I T T Y programme, the following list gives all the necessary interconnections:-

AFSK Board

IN1 - OUT on the USART board.
IN2 - 0V on the USART board.
OUT1 - microphone input on rig.
OUT2 - ground on rig.
+12V and 0V - from PSU.

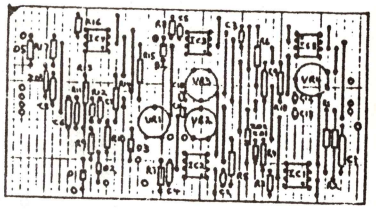
Receive Board

IN1 - external speaker or phones socket on rig.
IN2 - ground on rig.
OUT1 - IN on the USART board.
OUT2 - 0V on the USART board.
+12V, -12V and 0V - from PSU.

In Use

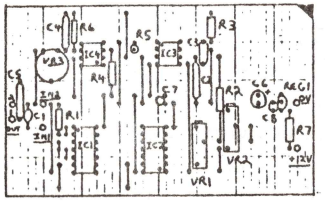
On receive the meter gives 3 peaks for each RTTY signal. 2 will give rapid fluctuations, but the centre one will give quite a smooth peak. After a little practice it is quite easy to tune in signals. Try and arrange your audio connections so that you can still hear the rig when the TU is connected. You will soon find that your ears provide a useful tuning aid! Only a low level of audio is needed to fully limit the input amplifier.
For receiving very weak signals an oscilloscope is very useful, but an explanation of its use is beyond the scope of these notes. For anyone wishing to use the technique, the output can be taken from points A and B on the board via 100k resistors.
The only adjustment needed on the AFSK board is to set VR3 to suit your rig, being careful not to overdrive it.

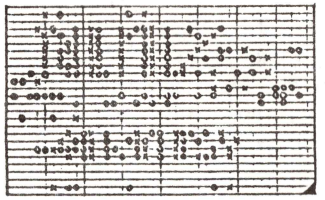
RECEIVE BOARD VEROBOARD LAYOUTS



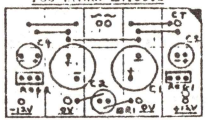
Component Side

AFSK BOARD VEROBOARD LAYOUTS

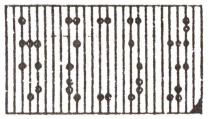


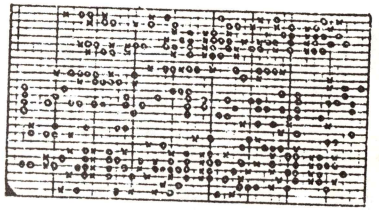
Component Side

Track Side

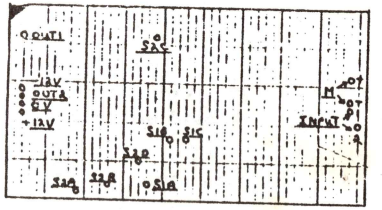
PSU BOARD LAYOUTS



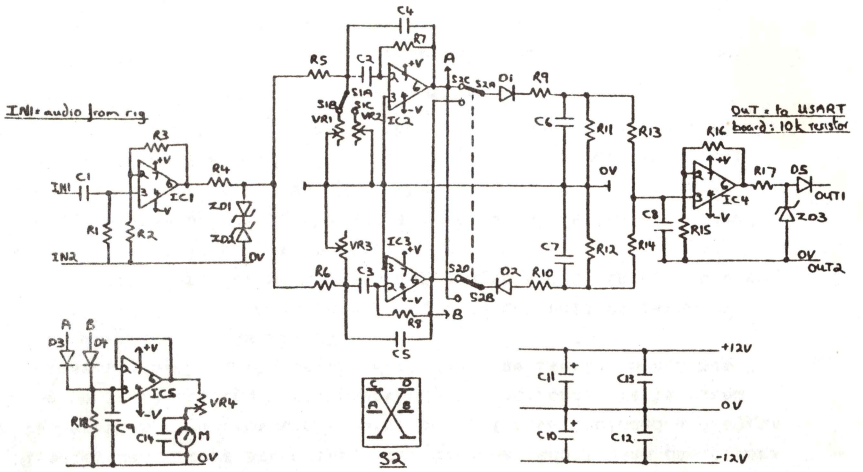
Component Side



Track Side

Track Side



Component Side - Veropins



S2

From R4
R5 or R6
100k preset
only if needed - see notes

Receive Board

RTTY Send Program & Interface for the Sinclair/Timex ZX81

By Brian Davis,* W9HLQ

One of the things in operating RTTY with non-electronic gear has been the nuisance of identifying. There is a lot of repetition such as the other station's call, my call, my name and QTH. I often envied those that had the new fancy "glass TTY" units that had all of that built in. Well it wasn't long after I got the Sinclair ZX81 that I realized that here was my answer. Following is an outline of a program that will send blocks or buffers of text at 60-wpm Baudot. Additionally I hope this will encourage others to dig into their Sinclair/Timex computers and put them to work on the ham bands.

The purpose of this article can be summarized:

1) Provide an automated way to assist in RTTY i-d sequences.

2) Use as a learning tool for building a simple interface for the ZX81.

3) Learn the connection between BASIC, machine language (ML) and the hardware.

4) Provide a basis for the higher-performance UART to be done later.

The method:

1) Build a simple two-TTL-chip interface with LEDs driven by a simple BASIC program. This will familiarize us with the system of entering ML programs into a REM statement and verifying that the interface is working. (This step is optional.)

2) Ennance the above program with the full ML TTY logic, add a relay to key the TTY loop and add the full BASIC support code.

A look at Fig. 1 will reveal the simplicity of the interface that we will be using. The LEDs are optional and for use in the preliminary checkouts. The system that I use is called "partial address decode" which means that only enough address lines are brought out of the interface to do the job. The address decode is done by a 74138 IC which is just made for our application. It is listed as a "1 of 8 decoder/demultiplexer." It will provide 8 different functions or signal lines when wired as shown. Note that due to a limitation of the Sinclair, only odd addresses are allowed. Even addresses will cause the computer to crash. I believe that this is due to the use of some addresses for video generation and sync. Anyway, all that we need is two for our application. The output of the 74138 consists of short pulses that, of course, are not suitable as is. The output is sent to a 7400 IC wired as an RS flip-flop. The output of the RS flip-flop (or latch, if you prefer) then drives LEDs and a relay via the 2N2222 NPN, or equivalent, transistor. The relay keys the TTY loop. An opto-isolator would be a natural application here, but the relay that I have in service has survived many dozens of QSOs.

The operation of this system is such that an "OUT 3" from the program will cause a pulse out of pin 12 of the 74138 to flip the 7400 RS flip-flop to the "on" state. An "OUT 5" will reverse the above sequence. The toggling of the 7400 will cause the transistor driver to cause the relay to pick up and drop. Therefore, the whole key to this system is to properly time the issuing of the OUT instructions. This is done in ML code that is called by BASIC. In this version, the ML code is kept to a minimum, and as much as possible is done in BASIC. This will aid in what is going on in the program. A later implementation could move some of the logic currently in BASIC to ML. This should speed up the sending rate which is currently about 30 wpm or about half speed. This is due to the relatively slow speed of BASIC.

The technique of using REM statements for storing ML programs is well documented in the available Sinclair literature (SYNC and SYNTAX), so I will not go into the reasoning for using this system except to say that I have found it to be the best system to isolate ML code. Those who check this program closely will note that the code in REM #1 is not used in this application. This ML code will be discussed later (another article) as it is part of my monitor program that will print your programs on your TTY machine. I chose to leave the REM in so that all of the addresses would match what I am using here. This will allow me to assist anyone who runs into trouble by comparing directly to my system. Also later, I bet that most of those who have the TTY machines will want to list their various programs on their TTY machines for hard copy. This will simplify doing it later. Anyway, the 1 REM will just be a "place holder" for this program.

Notes on the BASIC portion of the program:

Lines 1 - 3 (not shown with data as most of it is unprintable characters).

Line 108 requests input from the operator.

Line 110 - 230 check for special conditions and set up the shift.

Line 250 POKEs a bit configuration into a work byte for the ML program. This character is determined by using the internal code value of the character (see ZX81 manual) as an index value into a translation table that begins at 16616.

Line 260 is the call to the ML code.

Line 270 strips off the current character just sent off the string so that the next character can be sent.

Line 290 and lines 110 and 115 are for character counting to provide automatic CR/LF.

Instructions for Using this Prgram

1) Load the BASIC program as provided. The number of entries in the REMs is important. As you enter, a good idea might be to SAVE several copies as you progress. It is sure annoying to have to reenter a bunch of statements after a system crash!

2) Enter the translation table, 3 REM, from the table shown. It might be useful to verify it for accuracy by using the ML table provided and the program at "RUN 2000."

3) The TTY ML instructions can be entered also by typing "RUN 2100" and entering the proper values from Chart 1. Note that for now, noting has to be placed in the 1 REM except place holder

*17038 Oconto Avenue, Tinley Park, IL 60477.

"01234567890123..etc." The REMs each will have exactly the following number of entries. The number is critical since the system will be branching to this code as ML instructions.

1 REM -- 40 entries (can be any value since this isn't used)

2 REM -- 50 entries (ML instructions for send)

3 REM -- 65 entries (send tranlation table, this can be edited to vary your keyboard preferences)

4) SAVE several copies of your program before you attempt to run it. A USR call to ML may not return to BASIC if there is any error in the ML program. If the system is running (looping) in the ML portion, there is no way to BREAK and regain control of the system. Your only solution is to reset the system: power off and on again.

5) Edit lines 1xxx, the canned message area, for your needs.

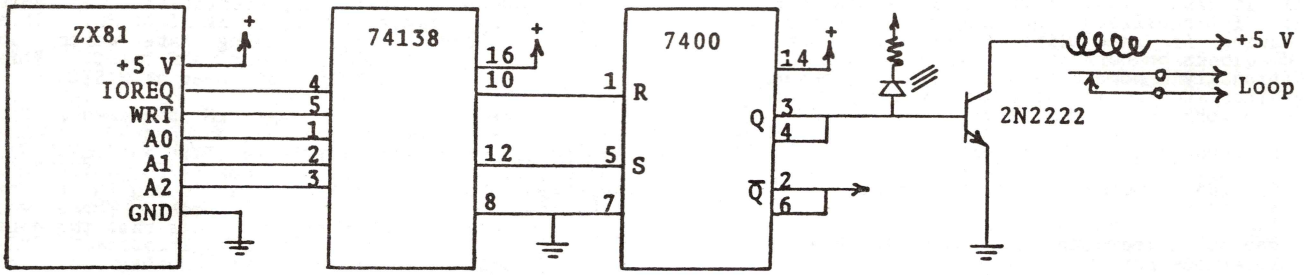
6) RUN. The prompt will direct you to enter several words or several lines of data. Press ENTER to send this text. After it has been sent, the prompt will reappear.

The following list of references will be handy if you are going to do any modifications to the ML portion of the program:

Zaks, "How to Program the Z80," Radio Shack.

Nichols, et. al., "Z80 Microprocessors, Programming, and Interfacing," Sams & Co.

Ciarcia, "Build Your Own Z80 Computer," Byte Books, McGraw-Hill.



I/O Address Decoder

R-S Flip-Flop Latch

RTTY Send Program Interface

Notes:

1. No UART is needed. Use simple circuit shown above.

2. The 74138 provides partial address decoding. It will provide signals for port addresses in the range of 0 thru 7; plenty for our small computer. Pin numbers shown will respond to OUT 3 and OUT 5 (port 3 and port 5). This corresponds to the BASIC and ML programs. Any two port addresses may be used. There are 4 OUT instructions that must correspond to output pins of the 74138.

3. The 7400 is wired as an R-S flip-flop. Any equivalent chip and circuit should work (it's going to be hard to beat the price of a 7400, though).

4. Wire loop to "NC" relay contacts so that loop is intact even if power is off.

5. Radio Shack's 275-216 relay works fine.

6. I found that removing the keying relay away from the computer kept loop keying transients out of the computer.

7. Some sort of transient suppression across the relay contacts will be necessary.

ZX81 RTTY Send Program

```
1 REM 012345678901234567...(40 ENTRIES-  NOT USED FOR RTTY SEND)
2 REM 012345678901234567...(50 ENTRIES-FOR RTTY SEND "ML" CODE)
3 REM 012345678901234567...(65 ENTRIES-FOR TRANSLATE TABLE DATA)
 10 REM ZX-81 "RTTY SEND PROGRAM"  (C)COPYRIGHT 1982 BRIAN DAVIS
 70 LET B$:" "
 80 LET T:0
 90 LET X:63
 95 REM -------CHECK FOR SPECIAL CONDITIONS/FUNCTIONS------
100 PRINT AT 10,1;"ENTER YOUR MESSAGE"
101 PRINT AT 12,3;":1 CALL AND MY ID"
102 PRINT AT 13,3;":2 MY ID ONLY"
103 PRINT AT 14,3;":3 CQ AND MY ID"
104 PRINT AT 15,3;":4 DATE"
105 PRINT AT 16,3;":6 K K K (OVER)"
106 PRINT AT 17,3;":8 CR/LF"
107 PRINT AT 18,3;":9 ENTER NEW CALL...."
108 INPUT A$
110 IF T>70 THEN GOSUB 400
115 IF T>63 AND CODE A$:0 THEN GOSUB 400
120 IF CODE A$:20 THEN GOTO 600
125 REM -----CHECK SHIFT OF CHARACTER-------
130 IF CODE A$38 OR CODE A$YR THEN GOTO 160
140 IF X:63 THEN GOTO 250
150 GOTO 210
160 IF CODE A$:0 OR X:59 THEN GOTO 250
190 LET X:59
200 GOTO 220
210 LET X:63
220 POKE 16561,X
230 LET U:USR 16560
250 POKE 16561,PEEK (16616#CODE A$)
260 LET U:USR 16560
270 LET A$:A$(2 TO )
280 IF A$:"" THEN GOTO 100
290 LET T:T#1
300 GOTO 110
399 REM ----------AUTO CR/LF ROUTINE------------
400 LET C$:"
410 POKE 16561,PEEK (16616#CODE C$)
420 LET U:USR 16560
430 LET C$:C$(2 TO )
440 IF NOT C$:"" THEN GOTO 410
450 LET T:0
460 RETURN
599 REM -------CANNED MESSAGE ROUTINE---------
600 LET A$:A$(2 TO )
610 LET L:(CODE A$)-28
620 IF L:0 THEN LIST 1040
630 GOTO 1000#L10
1010 LET A$:B$# ":2"
1012 GOSUB 400
1015 GOTO 110
1020 LET A$:"DE W9HLQ -BRIAN- IN TINLEY PARK, ILL."
1025 GOTO 110
1030 GOSUB 400
1032 LET A$:"CQ CQ CQ CQ CQ CQ DE W9HLQ W9HLQ W9HLQ "
1035 GOTO 110
1040 LET A$:"DATE-OCT.24, 1982 "
1045 GOTO 110
1060 LET A$:" K K K K"
1065 GOTO 110
1080 GOSUB 400
1082 LET A$:""
1085 GOTO 110
1090 PRINT "INPUT CALL"
1092 INPUT B$
1093 CLS
1094 GOTO 100
1999 REM ------TEMPORARY MAINTENANCE PGMS-----
2000 PRINT "ML LISTING"
2010 PRINT "INPUT START ADDRESS"
2020 INPUT A
2025 CLS
2030 FOR I:A TO A#10
2040 PRINT I;":";PEEK I
2050 NEXT I
2060 GOTO 2000
2100 PRINT "ML UPDATE"
2110 PRINT "INPUT START ADDRESS"
2120 INPUT A
2123 CLS
2125 PRINT "ENTER DATA (IN DECIMAL)"
2130 FOR I:A TO A#10
2153 PRINT I;":";
2155 INPUT D
2160 POKE I,D
2170 PRINT PEEK I
2180 NEXT I
2190 GOTO 2100
```

NOTES:

: IS AN EQUAL "=" SIGN
IS A PLUS "+" SIGN

QEX: The ARRL Experimenter's Exchange is published by the

American Radio Relay League
225 Main Street
Newington, CT 06111 USA
telephone 203-666-1541

Victor C. Clark, W4KFC
President

David Sumner, K1ZZ
General Manager

Paul L. Rinaldo, W4RI
Editor

Associate Editors:
David W. Borden, K8MMO (Data Communications)
Mark Forbes, KC9C (Components)
Geoffrey H. Krauss, WA2GFP (VHF+ Technology)

The purposes of QEX are to:

1) provide a medium for the exchange of ideas and information between Amateur Radio experimenters,

2) document advanced technical work in the Amateur Radio field, and

3) support efforts to advance the state of the Amateur Radio art.

Subscriptions are available to ARRL members and non-members at the rates shown on the QEX

ZX-81 RTTY TRANSMIT TRANSLATE TABLE TABLE 1 -USE WITH "3 REM"-
(THE "GRAPHIC CHAR" HAS THE REQUIRED BIT CONFIGURATION TO PRINT "PRINTED CHAR" ON THE TTY MACHINE. THE "DECIMAL VALUE" IS TO VERIFY VIA A "GOTO 2000" SINCE THE GRAPHICS CHARACTERS ARE SOMETIMES DIFFICULT TO EDIT.)

LOC.	DECIMAL VALUE	GRAPHIC CHAR.	ZX-81 KYBD CODE	PRINTED CHAR.
16616	132	:	0	SPACE
17	4	▘ (GR)	1	
18	59	V	2	
19	62	Y	3	
16620	61	X	4	
21	47	J	5	
22	49	L	6	
23	45	H	7	
24	192		8	
25	40	C	9	
26	50	N	10	
27	49	L	11	"
28	27	.	12	FIGS
29	41	D	13	$
16630	8		14	CR
31	57	T	15	?
32	47	J	16	(
33	50	N	17	)
34	31	3	18	LTRS
35	43	F	19	
36	46	L	20	: (=)
37	52	O	21	; (+)
38	131	G6	22	-
39	45	H	23	!
16640	61	X	24	/
41	2		25	LF
42	44	G	26	'
43	60	W	27	.
44	54	Q	28	0
45	55	R	29	1
46	51	N	30	2
47	1	G1	31	3
48	42	E	32	4
49	48	K	33	5
16650	53	P	34	6
51	39	B	35	7
52	38	A	36	8
53	56	S	37	9
54	131	G6	38	A
55	57	T	39	B
56	46	I	40	C
57	41	D	41	D
58	1	G1	42	E
59	45	H	43	F
16660	58	U	44	G
61	52	O	45	H
62	38	A	46	I
63	43	F	47	J
64	47	J	48	K
65	50	N	49	L
66	60	W	50	M
67	44	G	51	N
68	56	S	52	O
69	54	Q	53	P
16670	55	R	54	Q
71	42	E	55	R
72	5	SR	56	S
73	48	K	57	T
74	39	B	58	U
75	62	Y	59	V
76	51	N	60	W
77	61	X	61	X
78	53	P	62	Y
79	49	L	63	Z

S means shifted character

G means graphic character

```
RTTY TRANSMIT USING MACHINE LANGUAGE TIMING LOOPS(ZX-81|TS/1000)
     -BRIAN DAVIS, W9HLQ 5/19/82

402B 0E00      OUTPUT  LD    C,0     LOAD BIT PATTERN INTO REG C
402D 0606              LD    B,6     SET LOOP CONTROL
402F D303              OUT 3         SEND START BIT
4031 CD4B40            CALL DELAY    LET IT GO FOR A WHILE
4034 AF                XOR A         CLEAR THE CARRY FLAG
4035 CB39              SRL C         SHIFT RIGHT BIT INTO CARRY
4037 3004              JR NC,SP      NO CARRY=====>NO SPACE
4039 D305              OUT 5         SEND MARK
403B 1802              JR ENDCHR     JUMP AHEAD +2
403D D303          SP  OUT 3         SEND SPACE
403F 05        ENDCHR  DEC B         REDUCE LOOP COUNTER BY 1
4040 20EF              JR NZ,LOOP1   GO BACK IF ALL BITS NOT SENT (-17)
4042 D305              OUT 5         SEND STOP BIT
4044 CD4B40            CALL DELAY    DELAY FOR TWO DELAYS
4047 CD4B40            CALL DELAY
404A C9                RET           GO BACK TO BASIC
404B 1612       DELAY  LD D,18       SET OUTER LOOP COUNT
404D 1EF5       LOOP2  LD E,245      SET INNER LOOP COUNT
404F 1D         LOOP3  DEC E         COUNT DOWN BY 1
4050 20FD              JR NZ,LOOP3   INNER LOOP CONTROL ( GO BACK -3)
4052 15                DEC D         NOW THAT E IS 0 DEC D BY -1 AND GO BK
4053 20F8              JR NZ,LOOP2   OUTER LOOP CONTROL
4055 C9                RET           RETURN TO CALLER
```

Poke this is at 16560 (2REM) for the simple test interface.

```
16560  211  OUT 3     FLIP-ON
16561   03
16562  201  RETURN
16563  211  OUT 5     FLIP-OFF
16564   05
16565  201  RETURN
```

ZX-81 BTTY TRANSMIT MACHINE LANGUAGE
(ENTER DECIMAL VALUES INTO THE "2 REM" VIA ROUTINE AT "GOTO 2100")

"CHART 1"

```
ADDRESS HEX DECIMAL
        (ENTER THIS VALUE)
16560 0E 14 <-
16561 00 00
   62 06 06
   63 06 06
   64 D3 211
   65 03 03
   66 CD 205
   67 D0 208
   68 40 64
   69 AF 175
16570 CB 203
   71 39 57
   72 30 48
   73 04 04
   74 D3 211
   75 05 05
   76 18 24
   77 02 02
   78 D3 211
   79 03 03
16580 05 05
   81 20 32
   82 EF 239
   83 D3 211
   84 05 05
   85 CD 205
   86 D0 208
   87 40 64
   88 CD 205
   89 D0 208
16590 40 64
   91 C9 201
   92 16 22
   93 12 18 (ADJUST TO FINE TUNE PULSE TIMING)
   94 1E 30
   95 FE 245
   96 1D 29
   97 20 32
   98 FD 253
   99 15 21
16600 20 32
  601 F8 248
  602 C9 201
```

```
  1 REM 01234567890123456...(40 ENTRIES-NONOT USED FOR RTTY SEND)
  2 REM 01234567890123456...(50 ENTRIES-FOR RTTY SEND "ML" CODE)
  3 REM 01234567890123456...(50 ENTRIES-FOR TRANSLATE TABLE DATA)

 10 REM ZX-81 RTTY FLIP-FLOP TEST PROGRAM 1982 BRIAN DAVIS
100 PRINT "FLIP-FLOP TEST PROGRAM"
110 PRINT ";;;;"
120 PRINT "SELECT OPTION"
130 PRINT
140 PRINT "(1) REPEAT CHARACTER"
150 PRINT "(2) ONE TIME ONLY"
160 PRINT "(3) SLOW FLIP FLOP"
190 PAUSE 40000
200 LET I:CODE INKEY
210 POKE 16437,255
220 GOTO 2000#100(I-28)
260 LET U:USR 16560
270 LET A:A(2 TO )
280 IF A:"" THEN GOTO 3360
300 GOTO 110
2100 PRINT "INPUT THE CHARACTER YOU WANT REPEATED"
2110 INPUT A
2120 POKE 16561,CODE A
2130 LET U:USR 16560
2140 GOTO 2120
2200 CLS
2205 PRINT "INPUT YOUR CHAR"
2210 LET A:INKEY
2220 PAUSE 400
2230 POKE 16437,255
2240 POKE 16561,CODE A
2250 LET U:USR 16560
2260 GOTO 2200
2300 CLS
2310 PRINT "THIS IS THE SLOW FLIP-FLOP ROUTINE"
2320 REM THIS WILL -FLIP-
2340 LET U:USR 16560
2350 REM PAUSE FER A WHILE
2360 FOR I:1 TO 100
2362 NEXT I
2370 REM NOW -FLOP-
2380 LET U:USR 16563
2390 FOR I:1 TO 100
2400 NEXT I
2410 GOTO 2340
3000 PRINT "ML LISTING PGM"
3010 PRINT "INPUT START ADDRESS"
3020 INPUT L
3025 CLS
3030 FOR I:L TO L#10
3040 PRINT IV":";PEEK I
3050 NEXT I
3060 GOTO 9700
3100 PRINT "ML UPDATE PROGRAM"
3110 PRINT "INPUT START ADDRESS"
3120 INPUT L
3130 CLS
3140 FOR I:L TO L#10
3150 PRINT "INPUT VALUE "
3160 INPUT K
3170 POKE I,K
3180 PRINT IV":";PEEK I
3190 NEXT I
3200 GOTO 9800
```

FLIP-FLOP

Notes
: is an equal "=" sign
is a plus "+" sign

3040 PRINT I;"=";PEEK I

3180 PRINT I;"=";PEEK I

ADDR	DEC.											
16500	129	16501	144	16502	000	16503	000					
16504	000	16505	000	16506	000	16507	139	16508	065			
16509	000	"1" → 16510	001	16511	042	16512	000	16513	234			
16514	042	16515	123	16516	064	16517	205	16518	069			
16519	007	16520	042	16521	123	16522	064	16523	035			
16524	035	16525	078	16526	035	16527	070	16528	035			
16529	009	16530	034	16531	123	16532	064	16533	070			
16534	035	16535	078	16536	201	16537	042	16538	012			
16539	064	16540	035	16541	068	16542	077	16543	201			
16544	028	16545	029	16546	030	16547	031	16548	032			
16549	033	16550	034	"2" → 16551	035	16552	036	16553	037			
16554	118	16555	000	16556	002	16557	052	16558	000			
16559	234	16560	014	16561	055	16562	006	16563	006			
16564	211	16565	003	16566	205	16567	208	16568	064			
16569	175	16570	203	16571	057	16572	048	16573	004			
16574	211	16575	005	16576	024	16577	002	16578	211			
16579	003	16580	005	16581	032	16582	239	16583	211			
16584	005	16585	205	16586	208	16587	064	16588	205			
16589	208	16590	064	16591	201	16592	022	16593	018			
16594	030	16595	245	16596	029	16597	032	16598	253			
16599	021	16600	032	16601	248	16602	201	16603	031			
16604	032	16605	033	16606	034	"3" → 16607	035	16608	036			
16609	037	16610	118	16611	000	16612	003	16613	077			
16614	000	16615	234	16616	132	16617	004	16618	059			
16619	062	16620	061	16621	047	16622	049	16623	045			
16624	192	16625	040	16626	050	16627	049	16628	027			
16629	041	16630	008	16631	057	16632	047	16633	050			
16634	031	16635	027	16636	046	16637	052	16638	131			
16639	027	16640	061	16641	002	16642	044	16643	060			
16644	054	16645	055	16646	051	16647	001	16648	042			
16649	048	16650	053	16651	039	16652	038	16653	056			
16654	131	16655	057	16656	046	16657	041	16658	001			
16659	045	16660	058	16661	052	16662	038	16663	043			
16664	047	16665	050	16666	060	16667	044	16668	056			
16669	054	16670	055	16671	042	16672	005	16673	048			
16674	039	16675	062	16676	051	16677	061	16678	053			
16679	049	16680	061	16681	061	16682	008	16683	008			
16684	018	16685	018	16686	025	16687	018	16688	027			
16689	027	16690	027	16691	118	16692	000	16693	010			
16694	056	16695	000	16696	234	16697	063	16698	061			
16699	022	16700	036									

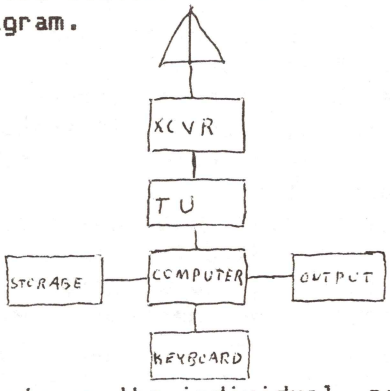
CHECK LIST - DECIMAL LISTING OF 1 REM, 2 REM, 3 REM

1 REM - ML FOR DISPLAY FILE (NOT USED HERE)
2 REM - RTTY SEND ML CODE
3 REM - RTTY XLATE TABLE (CONVERT SINCLAIR CODE TO TTY CODE)

THE BASIC RTTY STATION

Alex Burr

A block diagram for a basic RTTY computer controlled station would look like the following diagram.



A few comments on the individual parts are given here.

THE ANTENNA - no special attention need be paid to it for RTTY use.

THE XCEIVER - The transceiver (or separate receiver and transmitter) can be the usual station equipment. There are two main points to watch however. The RTTY signals are continuous signals. That is, the duty cycle is 100%. Your CW or SSB signal has a much lower duty cycle; so reduce tyour power input or make sure that your transmitter can handle a 100% duty cycle signal. The receiver must be able to handle the frequencies involved. The bandwidth is narrow but frequencies above khz are often present (see article on RTTY frequencies). The narrower your pass band is (while including the required frequencies) the better.

THE TU or terminal unit. Some clever programs can bypass this part of the system, but usually not very satisfactorily. This unit interfaces the transceiver to the computer. It produces the tones or signals for driving the transmitter and converts the receiver output tones to signals the computer can understand. It usually contains filters to reduce the effects of QRM. It can easily cost more that your computer or can be quite inexpensive. The expensive ones are more flexible, have better filters, and stable frequency shift keying arrangements. If a lot of high frequency (as distinct from VHF) work is planned, it is probably worth while to insist that the TU use both the mark and the space signal rather than just one of them as is possible. If both are used the bad effects of QRM and selective fading are reduced.

THE COMPUTER - The computer need not be elaborate. A ZX81, T/S1000, or T/S1500 will ke some of the programs mentioned in this ssue. The software is the key. If it is well designed, operating will be a pleasure.

OUTPUT - Some form of output is needed. Usually a TV or monitor is all that is needed. If a permanent record is wanted, a printer must be used.

STORAGE - Some data storage is needed for the program and for canned messages if you do not want to type them in each time you start up. Tape will work; EPOMS are better, if you can handle them.

KEYBOARD - RTTY goes faster if you can touch type although many have not learned. If you can, you probally will want an external keyboard or at leasdt a T/S1500. The ZX81 keyboard will work for the hunt and peck types.

CRYPTOGRAPHY SOFTWARE
FOR T/S-1000s with 2K or more

For Cryptosecure Tape Records
For Cryptosecure Communications
with Other T/S-1000s

Accepts Plain Text or Cipher
from Keyboard or Tape
Outputs Plain Text or Cipher
via TV Screen or Printer

Modem Use with Custom Mod
Record Length—320 characters
Double Encryption Yields
over 2 x 10^9 Combinations
Completely Debugged Listing
and Instructions $5.95

DEL PRODUCTS, INC.
13512 Keating St.
Rockville, MD 20853

RTTY LITERATURE

Alex Burr

Most hams, unless they are already active in RTTY, will need to read up on this operating mode before they know what equipment is required to operate. One of the best sources of information is described here in some detail. and some are mentioned at the end.

The Beginners RTTY Handbook published be the RTTY Journal (Box RY, Candiff-by-the Sea, CA 92007) at $8.00 is an excellent source of information for beginners. It has been around for quite some time so many of the circuits are a bit out of date, but the fundamentals have not changed. Furthermore the latest edition was updated in January 1982 so the sections on present practices are current too. The main objection that readers of this journal will have is the emphasis on mechanical RTTY terminals. But even this information will be useful for historical background.

The book itself is well printed with many diagrams and over 85 pages and about 150 topics mentioned (in no particular order) in the table of contents. Some of the topics I found of particular interest were "Start and Stop Pulses","Speed", "Types of Signals", "The Basic Receiving Process", "High vs Low Tones", "RTTY Operating Procedures", and "The Alphabet". Sinclair users can leave out all the parts about loop supplies, stunt boxes, mechanical modifications. and function levers. There is even a program which lets you play jingle bells on the Model 28. The pictures did not reproduce very well; but , because they are all of mecnanical terminals, that does not make any difference. The schematics and the printing are clear and easy to read.

For those really interested in RTTY, a subscription to the RTTY Journal at the above address would be a must (only $7.00 for a nicely printed publication).

Other sources of information include the ARRL Handbook (under specialized communications) and the November issue of CQ which is a special RTTY issue full of interesting articles. Also 73 Magazine (Peterborough,NM 03458) had a special RTTY issue in September 1982.They also published The New RTTY Handbook in 1978.

RTTY FREQUENCIES

Alex Burr

It is sometimes hard to keep all the frequencies used in RTTY work straigh There are four main types of frequencies; word frequencies, shift frequencies, mark and space frequencies, and transceiver frequencies.

The most important frequency is that at which the data is sent. The most common frequency is 60 words per minute. It is difficult for mechanical RTTY printers to change frequency and most amateur mechanical printers run at 60 wpm. With computers it is easy to change speed, but most use 60 wpm to be compatable with the mechanical printers. Some ASCII work is done at 100 wpm. Other rates used include 67, 75, and 150 wpm.

Most people know that a RTTY transmitter is continually on the air emitting one of two frequencies. These frequencies represent the zero and one of the five bit baudot binary code (given in another article). One frequency (mark) usually represents the zero and is the frequency transmitted when no information is being sent. The other frequency represents the one. The spacing between thse two frequencies is called the shift. There are three main shifts in use today. The usual shift for amateur work 170 Hz. Some Europeans amateur and a few U.S. stations use 850 Hz shift. Some commercial stations and military stations including MARS type nets use 450 Hz shifts.

Teletype signals leave the receiver as audio tones and the most common way to produce the transmitted frequencies is to modulate a SSB trnasmitter (on lower sideband) with the same two audio tones. Unfortunately there is not universal agreement on which two tones should be used. There are two main conventions as given in the table below.

TONE GROUP	MARK	170 HZ SPACE	850 HZ SPACE
LOW GROUP	1275	1445	2125
HIGH GROUP	2125	2295	2975

The high group tones are used by most U.S. amateurs. The low group tones are used by many Europeans.

The transmitter frequency is of little concern if you just want a contact. On 80m most RTTY signals are found between 3.60 and 3.65 MHz. On 20m look between 14.075 a 14.100 MHz. On 2m 146.7 MHz is often used and many areas have special 146.10/146.70 RTTY repeaters. However, if you want to set up a schedule, what frequency do you give? Remember, the signals are created by

modulating a carrier in a SSB tranceiver with two audio tones, and then surpressing the carrier and the upper sideband. Thus there are three obvious frequencies you could note: the surpressed carrier, the mark frequency, or the space frequency. (Just to make things more complicated the space frequency will be lower than the mark. Yes, I know that in the above table the space frequency is higher than the mark frequency, but we are using lower sideband, remember). In fact, the military and commercial interests, who are concerned with frequency management, use yet another frequency - the average of the mark and space frequencies. Highly recommended amateur practice is to use the mark frequency. However, that is not necessarily what your digital dial will read when you have the RTTY signal properly tuned in. Some read the surpressed carrier frequency, others the mark, and some the space frequency. Which varies from make to make and you must consult your owners manual or find out by experimenting.

CODE TABLE

Alex Burr

Here is a table which will be useful for those who are doing their own RTTY programing. As you know ASCII is a standard code for the exchange of information. Baudot is an earlier incompatable standard. The ZX81 and T/S1000 use a completely non-standard coding (the T/S1500 uses the same coding as the T/S1000).

Here is a discussion of the contents of the columns in the following table which contains ASCII, Baudot, and Timex/Sinclair information.

Col 1 Character
The ASCII character.

Col 2 Description
A word description of the ASCII character. (Note that the receiving computer does not have to be programed to act as the ASCII character implies.)

Col 3 Key
The keyboard key which, on an ASCII orientated conputer produces the indicated ASCII character. Note: The symbol ^ means the control key is pressed at the same time as the other key. Some computers use for ^ and some computers have dedicated keys that do not require the control key to be pressed.

Col 4 Decimal Code
The decimal code for the indicated character. Used with CHAR$ in a basic program. Note that because the 8th bit of the ASCII work was designed as an error check bit, there are only 128 possible ASCII characters (2^7 = 128). However, many computers have provision for using this 8th bit as a significant figure. The decimal, hex, and binary codes for these characters and the use to which these codes are put by the T/S computers is given in this table. These codes are easily distinguished by their initial 1. (Note that if the ASCII error checking plan is used, some ASCII codes will start with 1. Thus if parity type error checking is desired, code with decimal representation greater then 127 cannot be used.)

Col 5 Hex Code
The hex code for the indicated character.

Col 6 Binary Code
The binary code for the indicated character.

Col 7 Baudot Code
The Baudot code for the indicated character. The Baudot code has only 5 bits so can express only 32 (2^5 = 32) different characters, not enough for the letters and numbers. Hence the code FIGURES was devised to shift the receiving printer before certain characters were sent. These characters are indicated by a F following the 5 bit baudot word. The code LETTERS is usually sent to unshift the receiving printer. (Some printers automatically unshift on a LINE FEED).

Col 8 T/S1000 Code
Gives the use to which the indicated code is put by the ZX81, T/S1000, and T/S1500. Here gr means graphics character and the number gives the quadrant of the graphics character which is blacked. Quadrants are numbered clockwise from the upper left corner. The term grl identifies the half darkened quadrants.

Col 9 T/S2000 Code
Gives the use to which the indicated code is put by the Spectrum and the T/S2068. Here (lower case letter) indicates an address for a user created graphics symbol.

CHARACTER CODING

CHAR-ACTER	DESCRIPTION	KEY	DECIMAL CODE	HEX CODE	BINARY CODE	BAUDOT CODE	T/S 1000 CODE	T/S 2000 CODE
NUL	Null or tape feed	^@	0	00	00000000	00000	Space	not used
SOH	Start of heading	^A	1	01	00000001		gr1	not used
STX	Start of text	^B	2	02	00000010		gr2	not used
ETX	End of text	^C	3	03	00000011		gr1,2	not used
EOT	End of transmission	^D	4	04	00000100		gr4	not used
ENQ	Enquiry or WRU	^E	5	05	00000101	01001F	gr4,1	not used
ACK	Acknowledge or RU	^F	6	06	00000110		gr2,4	PRINT ,
BEL	Rings the bell	^G	7	07	00000111	01011F	gr1,2,4	EDIT
BS	Backspace or FEB	^H	8	08	00001000		gr11,2,3,4	cursor left
HT	Horizontal tab	^I	9	09	00001001		gr13,4	cursor right
LF	Line feed	^J	10	0A	00001010	00010	gr11,2	cursor down
VT	Vertical tab or VTAB	^K	11	0B	00001011		"	cursor up
FF	Form feed	^L	12	0C	00001100		pound sign	DELETE
CR	Carriage return	^M	13	0D	00001101	01000	$	ENTER
SO	Shift out	^N	14	0E	00001110		:	number (slug
SI	Shift in	^O	15	0F	00001111		?	not used
DLE	Data line escape	^P	16	10	00010000		(	INK control
DC1	Device control 1;(XON)	^Q	17	11	00010001		)	PAPER cntrl
DC2	Device control 2	^R	18	12	00010010		>	FLASH cntrl
DC3	Device control 3	^S	19	13	00010011		<	BRIGHT cntrl
DC4	Device cont'l 4;(XOFF)	^T	20	14	00010100		=	INVERSE ctrl
NAK	Negative acknowledge or ERR (error)	^U	21	15	00010101		+	OVER control
SYN	Synchronous idle	^V	22	16	00010110		-	AT control
ETB	End of transmission block	^W	23	17	00010111		*	TAB control
CAN	Cancel. Cancels current escape seq.	^X	24	18	00011000		/	not used
EM	End of medium	^Y	25	19	00011001		;	not used
SUB	Substitute	^Z	26	1A	00011010		'	not used
ESC	Escape	^[	27	1B	00011011		.	not used
FS	File separator	^\	28	1C	00011100		0	not used
GS	Group separator	^]	29	1D	00011101		1	not used
RS	Record separator	^^	30	1E	00011110		2	not used
US	Unit separator	^-	31	1F	00011111		3	not used
SP	Space	Space	32	20	00100000	00100	4	SPACE
!	Exclaimation point	!	33	21	00100001		5	!
"	Quotation mark	"	34	22	00100010	10001F	6	"
#	Number sign	#	35	23	00100011	10100F	7	#
$	Dollar sign	$	36	24	00100100	01101F	8	$
%	Percent sign	%	37	25	00100101		9	%
&	Ampersand	&	38	26	00100110	11010F	A	&
'	Apostrophe	'	39	27	00100111	00101F	B	'
(	Open parenthesis	(	40	28	00101000	01111F	C	(
)	Close parenthesis	)	41	29	00101001	10010F	D	)
*	Asterisk	*	42	2A	00101010		E	*
+	Plus sign	+	43	2B	00101011		F	+
,	Comma	,	44	2C	00101100	01100F	G	,
-	Hyphen or minus sign	-	45	2D	00101101	00011F	H	-
.	Period	.	46	2E	00101110	11100F	I	.
/	Slash	/	47	2F	00101111	11101F	J	/
0	Number 0	0	48	30	00110000	10110F	K	0
1	Number 1	1	49	31	00110001	10111F	L	1
2	Number 2	2	50	32	00110010	10011F	M	2
3	Number 3	3	51	33	00110011	00001F	N	3
4	Number 4	4	52	34	00110100	01010F	O	4
5	Number 5	5	53	35	00110101	10000F	P	5
6	Number 6	6	54	36	00110110	10101F	Q	6

CHARACTER CODING

CHAR-ACTER	DESCRIPTION	KEY	DECIMAL CODE	HEX CODE	BINARY CODE	BAUDOT CODE	T/S 1000 CODE	T/S 2000 CODE
7	Number 7	7	55	37	00110111	00111F	R	7
8	Number 8	8	56	38	00111000	00110F	S	8
9	Number 9	9	57	39	00111001	11000F	T	9
:	Colon	:	58	3A	00111010	01110F	U	:
;	Semicolon	;	59	3B	00111011	11110F	V	;
<	Less than	<	60	3C	00111100		W	<
=	Equal sign	=	61	3D	00111101		X	=
>	Greater than	>	62	3E	00111110		Y	>
?	Queation mark	?	63	3F	00111111	11001F	Z	?
@	At sign	@	64	40	01000000		RND	@
A	Letter A	A	65	41	01000001	00011	INKEYS$	A
B	Letter B	B	66	42	01000010	11001	PI	B
C	Letter C	C	67	43	01000011	01110	not used	C
D	Letter D	D	68	44	01000100	01001	not used	D
E	Letter E	E	69	45	01000101	00001	not used	E
F	Letter F	F	70	46	01000110	01101	not used	F
G	Letter G	G	71	47	01000111	11010	not used	G
H	Letter H	H	72	48	01001000	10100	not used	H
I	Letter I	I	73	49	01001001	00110	not used	I
J	Letter J	J	74	4A	01001010	01011	not used	J
K	Letter K	K	75	4B	01001011	01111	not used	K
L	Letter L	L	76	4C	01001100	10010	not used	L
M	Letter M	M	77	4D	01001101	11100	not used	M
N	Letter N	N	78	4E	01001110	01100	not used	N
O	Letter O	O	79	4F	01001111	11000	not used	O
P	Letter P	P	80	50	01010000	10110	not used	P
Q	Letter Q	Q	81	51	01010001	10111	not used	Q
R	Letter R	R	82	52	01010010	01010	not used	R
S	Letter S	S	83	53	01010011	00101	not used	S
T	Letter T	T	84	54	01010100	10000	not used	T
U	Letter U	U	85	55	01010101	00111	not used	U
V	Letter V	V	86	56	01010110	11110	not used	V
W	Letter W	W	87	57	01010111	10011	not used	W
X	Letter X	X	88	58	01011000	11101	not used	X
Y	Letter Y	Y	89	59	01011001	10101	not used	Y
Z	Letter Z	Z	90	5A	01011010	10001	not used	Z
[	Open brackets	[	91	5B	01011011		not used	[
\	Reverse slash	\	92	5C	01011100		not used	\ ??
]	Close brackets	]	93	5D	01011101		not used	]
^	Caret or up arrow	^	94	5E	01011110		not used	^
_	Underscore	_	95	5F	01011111		not used	_
`	Grave accemt	^= *	96	60	01100000		not used	pound sign
a	Letter a	a	97	61	01100001		not used	a
b	Letter b	b	98	62	01100010		not used	b
c	Letter c	c	99	63	01100011		not used	c
d	Letter d	d	100	64	01100100		not used	d
e	Letter e	e	101	65	01100101		not used	e
f	Letter f	f	102	66	01100110		not used	f
g	Letter g	g	103	67	01100111		not used	g
h	Letter h	h	104	68	01101000		not used	h
i	Letter i	i	105	69	01101001		not used	i
j	Letter j	j	106	6A	01101010		not used	j
k	Letter k	k	107	6B	01101011		not used	k
l	Letter l	l	108	6C	01101100		not used	l
m	Letter m	m	109	6D	01101101		not used	m
n	Letter n	n	110	6E	01101110		not used	n
o	Letter o	o	111	6F	01101111		not used	o
p	Letter p	p	112	70	01110000		cursor up	p
q	Letter q	q	113	71	01110001		cursor down	q
r	Letter r	r	114	72	01110010		cursor left	r
s	Letter s	s	115	73	01110011		cursor rights	
t	Letter t	t	116	74	01110100		GRAPHICS	t
u	Letter u	u	117	75	01110101		EDIT	u

CHARACTER CODING

CHARACTER	DESCRIPTION	KEY		DECIMAL CODE	HEX CODE	BINARY CODE	BAUDOT CODE	T/S 1000 CODE	T/S 2000 CODE
v	Letter v	v		118	76	01110110		ENTER	v
w	Letter w	w		119	77	01110111		DELETE	w
x	Letter x	x		120	78	01111000		K/L mode	x
y	Letter y	y		121	79	01111001		FUNCTION	y
z	Letter z	z		122	7A	01111010		not used	z
{	Left brace	^(	*	123	7B	01111011		not used	^((ONERR)
¦	Vertical bar	¦		124	7C	01111100		not used	STICK
}	Right brace	^)	*	125	7D	01111101		not used	^) (SOUND)
~	Tilde	^/	*	126	7E	01111110		number	FREE
DEL	Delete	^?	*	127	7F	01111111		cursor	(c) (RESET)
	FIGURES						11011		
	LETTERS						11111		

NOTE: The above two items are used in TTY only.
An F in the Bardot column means that a FIGURES code must have preceeded that code. The F codes have been taken from the Telex variation which is almost identical to the International Alphabet #2.

CHARACTER CODING

DECIMAL CODE	HEX CODE	BINARY CODE	T/S 1000 CODE	T/S 2000 CODE
128	80	10000000	gr1,2,3,4	gr white
129	81	10000001	gr2,3,4	gr2
130	82	10000010	gr1,3,4	gr1
131	83	10000011	gr3,4	gr1,2
132	84	10000100	gr1,2,3	gr3
133	85	10000101	gr2,3	gr2,3
134	86	10000110	gr1,3	gr1,3
135	87	10000111	gr3	gr1,2,3
136	88	10001000	grl1,2,3,4	gr4
137	89	10001001	gr1,2,l3,4	gr2,4
138	8A	10001010	grl1,2,d3,4	gr4,1
139	8B	10001011	inverse "	gr1,2,4
140	8C	10001100	inverse	gr3,4
141	8D	10001101	inverse $	gr2,3,4
142	8E	10001110	inverse :	gr1,3,4
143	8F	10001111	inverse ?	gr1,2,3,4
144	90	10010000	inverse (	(a) user
145	91	10010001	inverse)	(b) graphics
146	92	10010010	inverse)	(c)
147	93	10010011	inverse (	(d)
148	94	10010100	inverse =	(e)
149	95	10010101	inverse +	(f)
150	96	10010110	inverse -	(g)
151	97	10010111	inverse *	(h)
152	98	10011001	inverse /	(i)
153	99	10011001	inverse ;	(j)
154	9A	10011010	inverse ,	(k)
155	9B	10011011	inverse .	(l)
156	9C	10011100	inverse 0	(m)
157	9D	10011101	inverse 1	(n)
158	9E	10011110	inverse 2	(o)
159	9F	10011111	inverse 3	(p)
160	A0	10100000	inverse 4	(q)
161	A1	10100001	inverse 5	(r)
162	A2	10100010	inverse 6	(s)
163	A3	10100011	inverse 7	(t)
164	A4	10100100	inverse 8	(u)
165	A5	10100101	inverse 9	RND
166	A6	10100110	inverse A	INKEY$
167	A7	10100111	inverse B	PI
168	A8	10101000	inverse C	FN
169	A9	10101001	inverse D	POINT
170	AA	10101010	inverse E	SCREEN$
171	AB	10101011	inverse F	ATTR
172	AC	10101100	inverse G	AT
173	AD	10101101	inverse H	TAB
174	AE	10101110	inverse I	VAL$
175	AF	10101111	inverse J	CODE
176	B0	10110000	inverse K	VAL
177	B1	10110001	inverse L	LEN
178	B2	10110010	inverse M	SIN
179	B3	10110011	inverse N	COS
180	B4	10110100	inverse O	TAN
181	B5	10110101	inverse P	ASN
182	B6	10110110	inverse Q	ACS

CHARACTER CODING

DECIMAL CODE	HEX CODE	BINARY CODE	T/S 1000 CODE	T/S 2000 CODE
183	B7	10110111	inverse R	ATN
184	B8	10111000	inverse S	LN
185	B9	10111001	inverse T	EXP
186	BA	10111010	inverse U	INT
187	BB	10111011	inverse V	SQR
188	BC	10111100	inverse W	SGN
189	BD	10111101	inverse X	ABS
190	BE	10111110	inverse Y	PEEK
191	BF	10111111	inverse Z	IN
192	C0	11000000	""	USER
193	C1	11000001	AT	STR$
194	C2	11000010	TAB	CHR$
195	C3	11000011	not used	NOT
196	C4	11000100	CODE	BIN
197	C5	11000101	VAL	OR
198	C6	11000110	LEN	AND
199	C7	11000111	SIN	<=
200	C8	11001000	COS	>=
201	C9	11001001	TAN	<>
202	CA	11001010	ASN	LINE
203	CB	11001011	ACS	THEN
204	CC	11001100	ATN	TO
205	CD	11001101	LN	STEP
206	CE	11001110	EXP	DEF FN
207	CF	11001111	INT	CAT
208	D0	11010000	SQR	FORMAT
209	D1	11010001	SGN	MOVE
210	D2	11010010	ABS	ERASE
211	D3	11010011	PEEK	OPEN#
212	D4	11010100	USR	CLOSE#
213	D5	11010101	STR$	MERGE
214	D6	11010110	CHR$	VERIFY
215	D7	11010111	NOT	BEEP
216	D8	11011000	**	CIRCLE
217	D9	11011001	OR	INK
218	DA	11011010	AND	PAPER
219	DB	11011011	<=	FLASH
220	DC	11011100	>=	BRIGHT
221	DD	11011101	<>	INVERSE
222	DE	11011110	THEN	OVER
223	DF	11011111	TO	OUT
224	E0	11100000	STEP	LPRINT
225	E1	11100001	LPRINT	LLIST
226	E2	11100010	LLIST	STOP
227	E3	11100011	STOP	READ
228	E4	11100100	SLOW	DATA
229	E5	11100101	FAST	RESTORE
230	E6	11100110	NEW	NEW
231	E7	11100111	SCROLL	BOARDER
232	E8	11101000	CONT	CONTINUE
233	E9	11101001	DIM	DIM
234	EA	11101010	REM	REM
235	EB	11101011	FOR	FOR
236	EC	11101100	GOTO	GOTO
237	ED	11101101	GOSUB	GOSUB
238	EE	11101110	INPUT	INPUT
239	EF	11101111	LOAD	LOAD
240	F0	11110000	LIST	LIST
241	F1	11110001	LET	LET
242	F2	11110010	PAUSE	PAUSE
243	F3	11110011	NEXT	NEXT
244	F4	11110100	POLK	POLK
245	F5	11110101	PRINT	PRINT

CHARACTER CODING

DECIMAL CODE	HEX CODE	BINARY CODE	T/S 1000 CODE	T/S 2000 CODE
246	F6	11110110	PLOT	PLOT
247	F7	11110111	RUN	RUN
248	F8	11111000	SAVE	SAVE
249	F9	11111001	RAND	RANDOMIZE
250	FA	11111010	IF	IF
251	FB	11111011	CLS	CLS
252	FC	11111100	UNPLOT	DRAW
253	FD	11111101	CLEAR	CLEAR
254	FE	11111110	RETURN	RETURN
255	FF	11111111	COPY	COPY

NEW!

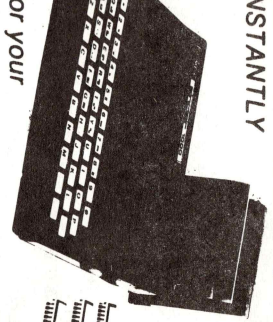
ROMPAK PLUG IN

PROGRAM CARTRIDGES

Featuring software that runs INSTANTLY

TS/1000 TS/1500

- ELIMINATE CASSETTE loading time and frustration! (No tapes, no cables, no gadgets.)
- Get the LOW COST cartridge and pop in any solid state Eprom program. (see selection below.)
- Quickly change Eprom programs anytime with the Z.I.F. "pop-it" socket.
- Lets you EXPAND with the built-in piggyback port for your rampack,printer etc.
- Buy software the SAFE way - ORDER YOUR CARTRIDGE TODAY!



$16.95

POP-IN EPROMS

ENTERTAINMENT

- ACTION PAK - Two super fast arcade games. . .$15.95.
- PARTY PAK - Two fun games for the crowd. . .$15.95.
- BLACK JACK PAK - Break the bank at Vegas. . .$14.95.
- MUSE PAK - Composes poetry! Changeable vocabulary . . .$14.95.

16K required

PRACTICAL

- MATH PAK - Graphing, curve fitting, matrix solving. . .$14.95.
- FINANCIAL PAK - Loans, savings, investment analyses. . . 15.95.
- STAT PAK - Advanced statistics with plotting. Complete. . . $16.95.

UTILITY

- HEX PAK - Great machine code monitor/ editor. . .$13.95.
- WATCH FOR - TOOL PAK and GRAPHIC PAK coming soon.

SPECIAL OFFER!
- GALACTIC PAK Fast machine code arcade game $9.95

ORDER FORM

☐ CARTRIDGE $16.95
(Accepts 2764 & 2732 Eproms)
☐ CARTRIDGE KIT ..... $ 9.95
(6 piece kit - regular socket only)
POP-IN EPROMS
"_____PAK" @ $ ..... EA
"_____PAK" @ $ ..... EA
"_____PAK" @ $ ..... EA
Cal. res. add 6½% Shipping add $2.00
Total .....

Name _____

Address _____

Send check or money order to:

Rompak, Inc. Suite 100
8206 Blackburn Avenue
Los Angeles California 90048

Editor's Note: Programs are either 'cut & pasted' and put with the heading where practical, or will appear at the nearest convenient insertion point.

ZX81/Baudot TTY Interface & Program
Dallas Williams, WA0MRG

Here's a nice article telling how to interface an ol' Teletype machine to the Sinclair. Be sure to follow his tips on using an optocoupler, the voltages in a teletype would fry the Sinclair!

ZX81 RTTY System
Henning Andresen, OZ1BII

From the British Sinclair Amateur Radio User Group Newsletter done by Paul Newman, G4INP. (Reprinted with permission). Here's a nice system, complete with TU, AFSK, and interface. We are only printing the ZX81 program, send SASE (to W4GHV) if interested in the Spectrum version. A PC board is available from G4INP. The British price is 6 lbs, 50 p (whatever that means), write him for US prices. Note that OZ1BII's schematic is copyrighted, which means you can build your own for experimenting, but you cannot (legally) make them to sell to others.

RTTY Terminal Unit
Roger Barker, G4IDE

Here's another article from Paul's newsletter. Send a SASE (to W4GHV) for a darker copy of this article if it doesn't print well.

ZX81 RTTY Send Program & Interface
Brian Davis, W9HLQ

Here's a reprint of an excellent article that appeared in the Dec 82 issue of QEX (not QZX)! If you haven't seen an issue of QEX, you should send for one, its a nice magazine. (I wonder if a companion receive program has been developed for this?)

Editor's Doings
Bo Barry, W4GHV

How many of you bought your Sinclair with intentions of putting it on RTTY? How many have successfully accomplished the task? Hopefully this issue will inspire the procrastinators into getting on RTTY, it is a lot of fun. I was first exposed to it in 1969, working a VK on my very first CQ! I became a Model 15 expert in short order. Unfortunately, my RTTY activity has been intermittent and I am a bit behind the times. I would thoroughly enjoy an article from our readers describing what's going on, what frequencies to listen to, etc. I sure need a good CONSTANT signal to use for troubleshooting, ham QSO's don't last long enough!

Following are ideas & opinions of several RTTY AFSK terminal units that I have built or am building. They may be of interest to those of you lookig for an easy way to get on RTTY. First, the one that I built back in 1969, which I successfully revived this week, is from an article by WA6JMM in July '69 73 Magazine. While no PC layout is given, it is rathe simple, using an op amp and a voltage comparator. It uses two 88mH telephone toroids and has a FM ratio-detector circuit, making it relatively insensitive to amplitude variations due to signal fading. I plan to modify it to provide a TTL output for use with my Sinclair RTTY setup. I added some toroid bandpass filters from a QST article and was well pleased with the performance on narrow and wide shift. The tuned LC circuits are not easily adjusted for various shifts.

Another one that I am presently building is by KB9AT in Oct '78 Ham Radio Magazine. Called the digiratt PLL_2, it uses two phase-locked-loop ICs to detect both the mark & space and has a multiplexer IC to switch between the mark & space outputs. This allows copy when one of the two signals is missing due to interference or fading. The PLLs use simple trimpots for tuning, allowing setup for several shifts. Also included is an active bandpass filter and LEDs to aid in tuning. I thought I was familiar with most families of ICs, but the MC1808, 9946, & 9309 ICs used in this threw me for a loop. I found TTL subs for two of them, but landed up buying the 9309. This is a simple one to build and a PC board may be available from Circuit Board Specialists. I can't testify (yet) as to how well this TU works, but I hope to have it finished soon. If enough people were interested in it, I could have a PC negative made and could loan it to those of you set up for making PC boards.

The last one that I am thinking about,

is by WB2BWJ and was in Dec '80 QST. It does both AFSK and demodulation with only 2 ICs (the Exar 2206 & 2211). I am feverishly looking through my junk for an AFSK I built a while back, using either the 2206 or the 8038 waveform generator IC. It worked fine and I liked the 'phase-continuous' feature, where the transition from mark to space occurs during the zero crossing of the waveform. This results in a much cleaner sine wave output and less worry about harmonics when hooked to a SSB XMTR. If I don't find mine I will land up building this one. I plan to get a negative made, in case. I think this would make a simple modem for computer use also. The PLL only activates on a mark tone and I have no idea as to how well this simple TU would perform. A PC board is (in 1980) available from WB2BWJ.

As for direct FSK of CW transmitters, a simple circuit of a few resistors and a varicap diode across the VFO works fine. I have lost the one I came up with for my old HW-100 XCVR years ago.

That's about all I know about RTTY, if anyone has interest in any of the above circuits send a SASE and I will gladly send info on them. We would also welcome articles on RTTY. Tuning indicators and active bandpass filters are two interesting topics.

Lastly, here is a list of RTTY related articles that have already appeared in QZX, for the new subscribers. Unfortunately, we are temporarily out of back issues!

ZX81 RTTY Receive Program, WA0MRG, Mar 83 QZX. (Only a TU is required)

Simple RTTY TU Circuit, WA0MRG, Apr 83 QZX.

ZX81 RTTY Transmit & Receive Program with hardware interface, AF8J & WB8UJB. (TU & AFSK required)

Active Bandpass Filter Program, WB4FXD, Jun 83 QZX.

P.S. Here are a few 'design tips' or comments on RTTY received from Terry, N8BIF. He has tried several TU's using one 567 PLL, two 567's, 565's, and others. The only one giving success used LC filters (like my 1969 TU) and the Exar 2211 IC. Watch out for excessive audio input levels, it affects the IC's bandwidth. He used the 8038 function generator IC for AFSK. Thanks for the tips Terry.

TS-1500 second impressions: I have had but a little time to play with it, and am already a bit confused. The free game that came with my sub to the Timex/Sinclair Users Mag that has a fast loader refuses to load. Another program I have, loads but the machine language portion bombs. I thought all the software was compatible? I plan to chase down a schematic & details and to disassemble both the ZX81 & 1500 ROMs to see what's going on. The 1500 finally appeared on the local scene (@$69). The color model (@$158) is outselling the 1500, contrary to my predictions. They put it on sale for $139 and sold out. Alex has a rain check and perhaps he can tell us about it later. The serial # on the 1500 in the store was only 2600, so the boat from Portugal is still running slow. Your findings on the 1500 differences would be greatly appreciated.

Products in Review

Here's an excellent review of the AF8J RTTY interface/program done by Terry, N8BIF.

In the April 1983 issue of QZX, Larry Willson presented his RTTY program. If you were like me, the prospect of rtty made normal life impossible. That version of rtty was admittedly primative; after a full screen on either transmit or receive the program stopped. Never the less, it did work.

When his "more sophisticated" RTTY program arrived, I immediately tried it. Even though the interface for the first was different than that for the new version, the "more sophisticated" program ran without problem.

With the cassette came a single sheet of explanation and another with the new interface schematic. The new interface is a simplified version of its predecessor, containing only 4 I.C.s and a very few other components, all of which are available from Radio Shack.

The explanation sheet was short but sufficiant. It did contain the statement that the program would run in two k of ram, no more or less and the fact of the matter is that the program will run with the 16k versio as well by setting ramtop to 18432 after it is running. The display

file characteristics change below. 3.3K of ram and I think that the AF8J RTTY program takes advantage of these differences.

Basic operation is really quite simple. The program loads in about 30 seconds and is self-running; initially, in the receive mode. The first things that you see on the screen is a copyright banner at the top of the screen and at the bottom is the mode monitor. The "constant mode monitor" consists of a line of inverse characters telling you that you are in receive or transmit. This may sound like just another bell or whistle, but it really is a nice feature, especially if you are like me and forget to return the computer to the receive mode (by hitting the equals key) and are sitting there listening to the other guys tones and wondering why there is no printing.

In operation, the copy prints on a line near the bottom of the screen and then scrolls up so that you never run out of screen. You won't usually get a full line though, I think that the program tries to scroll during a space in order to prevent parts of the same word ending up on two lines. This takes place near the end of a line and the overall effect is quite professional.

I've had several QSO's with this program and the only time that I lost a letter, the fault was with either QRM, receiver tuning, and other assorted forms of operator error. The program prints all standard Baudot characters with the exception of #, &, !, and the 'bell'. The apostrophe is sent and received as the semicolon (;). Also, as I said, to switch from transmit to receive, you just type the equals sign.

Speaking of transmitting, to start transmitting just hit any key and then start typing. Just remember that the first key that you hit is not sent, that is it causes the program to switch from the receive to the transmit and then is lost. Initially, I considered this to be a drawback but after operating the program for awhile, I find that it is not bothersome and that it does prevent false keying.

As for the transmit speed, I could not out-type it with my three-fingered typing except when I switched from "figures" to "letters" or the other way around. The program monitors whether you are in upper case or lower case and when it senses a change, it transmits automatically the correct shift. For example, when you type " M " and then " 9 " the program would transmit first the code for M and then the code for figures and finally that for nine.

By sticking in the "extra" machine cycle, you can get ahead of the machine. If you are a touch typist, you should be aware of this momentary "increase in speed".

The two features missing from this program are type ahead buffer and a set of message buffers for transmitting canned messages. Neither of these are needed but are added conveniences.

The AF8J RTTY program is an inexpensive way to a very effective rtty operation. For the price of a T-S 1000, the interface and homebrew modem, and the software, getting into radio teletype could cost you less than $100.

A couple of weeks ago I received a mimeographed flyer from Larry saying that he had an ASCII program that would run using the same interface, changing only one wire and that on the same that on the same tape would be another RTTY to work on the new interface. Both the RTTY and the ASCII programs would have word wraparound and would retail for $15.

All in all, I am impressed and pleased with the program and the interface. It's just plain fun even if it doesn't correct my spelling.

Next, here's details on NU4V's new RTTY program and interface. Alex and I were supposed to have ours going before press time, but we haven't quite finished. Called NRAP (NU4V's RTTY ASCII Program), this program is loaded with features, and the simple interface board which uses the super slick INS8250 chip can be easily interfaced to any terminal unit having a TTL output. Following are some of its features:

* Combined receive & type ahead buffer space of over 9000 characters with a 16K machine. Their sizes can be easily changed by the user.

* Baud rates are selectable from the keyboard, ranging from 45 to 100 for Baudot and 110,150, & 300 baud for ASCII.
* Ten message buffers (254 characters each) can be created and can be reviewed or changed while receiving a QSO.
* The type ahead buffer can be created while receiving, and can be retransmitted if desired.
* The display consists of 16 lines of receive data and four lines of type ahead data. Both displays scroll, but are not lost until cleared.
* It uses 'unshift on space' and has a keyboard 'change shift' command.
* It allows hard copy in real time with a full size printer or after the QSO with the Sinclair printer.

It has four main modes of operation are: Receive, Transmit, Message Enter/Review, and Receive Buffer Review. NRAP is fairly easy to use, for example to increase the baud rate, hit the UP arrow (shift 7). Hit shift A for ASCII or shift R for RTTY (Baudot). The canned messages stored in the buffers can be easily incorportated into the text you type into the type ahead buffer, being identified by a shift 1 followed by the buffer number. And here's the neatest feature! It transmits the words a word at a time. In other words (a play on words), you can screw up a word, use the delete key to clean it up and it is finally sent when you hit the next space key. The guy on the other end thinks you are thinking during the long pause, while you are really trying to spell your name right! If you are able to type and think faster than the 'other end' you can, of course type into the type ahead buffer while he is transmitting to you, and then your response clips along at full speed, having him thinking you really know how to type! And to think I used to get excited about using paper tape on my ol' Model 15! (This ol' editor has gotten the RTTY bug again, and is going to be on by XMAS!)

I hope I have done NRAP justice, the instructions are quite complete, and we haven't actually got to use it yet. One note on the PC board, its orientation is reversed compared to the CWSS interface he made earlier. The components go to the rear and the edge connector is soldered to the foil side of the board.

That's about it, write to Cliff Nunnery, NU4V, 313 Vaughn St., Ft. "Walton Beach, Fla., 32548 if interested. The program sells for $15 and a blank PC board (with a small extender board) sells for $14.

UNCLASSIFIED ADS

WANTED: UNCLASSIFIED ADS for publication in QZX. Cost is 1 cent for each keystroke (characters, blanks, etc.). Payment must be included with order. This is an excellent way to sell your program or unwanted equipment, ask for help, or find a specific program. Send your ad in with payment to QZX, 2025 O'Donnell, Las Cruces, NM 88001.

FOR SALE: ZX81/TS1000 edgeboard wire-wrap connectors; keyed; bifurcated contacts; manufactured by TI; $5.00 each. Pc piggy-back extender boards; $1.50 each. Economical experimenting with Radio-Shack's 276-158 and 161 project boards and these connectors. Add $0.75 shipping per order. Paul Hunter, 1630 Forest Hills Drive, Okemos, MI 48864.

FOR SALE: TS1000 HAM HELPER PROGRAMS - Some written for county hunting and all are in BASIC for easy mods to suit individual taste. Write for details (SASE appreciated) to KA2LHO, Kraig D. Pritts, Box 198, Leonardsville, NY 13364 (see QZX for sample).

FOR SALE: Surplus keyboards suitable for T/S 1000. $30.00 including postage. Alex. F. Burr, 2025 O'Donnell, Las Cruces, NM 88001.

FOR SALE: ASCII/RTTY transceiver program. The entire program is written in machine code. It fits in just 2K of RAM. Features: word wrap, auto scroll, unshift on space, auto Fig/Ltrs., auto CR/LF, key transmitter from keyboard, and on screen mode reminder. Program and interface schematic $15.00. Silk screened and tinned interface board with tone demodulator and tone generator ability, $15.00. Larry Willson, 149 Hoehn Ct. P.O. Box 465, Dimondale, MI. 48821

QZX FRIENDS

KT5V -- Dave has a T/S 1000, is 21 years old, and has been licensed for 8 years. He is planning to add NU4V cw. He also liked the J[illegible]g double issue which he received dispite the fact that we messed up his address.

WA4SFS -- H. D. has a T/S 1000 with 64K and a big keyboard.

WB0QPI -- LLoyd has just moved to Grand Junction, Colorado, and is concentrating on learning to use his ZX81.

KB8WD -- Lloyd uses his ZX81 with 16K RAM and printer on cw and RTTY.

WB3JDP -- Hal is looking for info on modifying surplus keyboards.

VE3MJY -- Phil has a ZX81 and wishes to develop it for RTTY, CW, and satalite antenna tracking.

WB9NRK -- Dale received his T/S for Christmas and is having a ball with it using the NU4V cw program with Drake Twins. He has made two net control lists, one for a local 2 meter net and the other for an ARES county net.

KA5ETD -- John has just purchased a Suntronics keyboard for his 1000 and finds it a nice unit and inexpensive.

W1EFK -- C. L. sent us a nice QSL card which he had his 2040 printer produce which included a very good drawing of his state. H[illegible] also is using his T/S 1000 to send and receive cw.

WB1DRE -- Tony is one of the original subscribers who has rejoined. He has a ZX81, 16K RAM, and a model 28 RTTY machine.

KA4FRF -- Tom writes from Karachi that he really looks forward to receiving QZX each month. He also suggests that we publish the addresses of our authors, which we will try to do.

K4BRX -- R.F. says that he will put some of the QZX programs to work as soon as he gets a Sam's info sheet to repair his B&W TV set so he can use it as a monitor.

WB3JDP -- Hal is retired and has lots of time for his rig and T/S 1000. He has added an external keyboard from BNF.

N8BIF -- Terry writes "I bought a kit ZX-81 in April of 1982, adding first a keyboard and then adding 4K of internal RAM followed by 16K RAM and the Sinclair printer. I do as much of my programming in machine code as possible for speed and size but I also play around with some FORTH. I guess that my favorite accomplishment is a 7 function toolkit that resides in less than 1K with the following subroutines: memory remaining; checksum; delete; remkill; find$; and renumber."

WA4PSW -- Forrest has purchased a Hunter memory boaed and installed the ZXLR-8 program in it at location 14000 for ease of remembering. He heartily recommends this combination.

WIBYY -- Ted has installed his ZX81 in a Suntronics unit with sucess. He also thinks that NU4V's cw system is swell.

BITS AND BYTES

28. OE2ZDC had an item for those with the Sight and Sound Video Reverser installed on their ZX orT/S computers. Erratic display problems can come from poor connections at the video output of the add-on board. The stiff wires as supplied easily breaK loose from the board and pull up the traces. He had to completely replace one broken trace. Use light stranded wire such as 24 guage speaker cable instead of the stiff wire suplied and you will save trouble later on.

29. OE2ZDC also said the excellent keyboard from Double H Electronics in San Antonio, TX comes with ribbon cable to be soldered to the ZX or T/S and to the keyboard itself. It is much easier and better to make use of the female connector already attached to the keyboard. Do not solder, just make up two male plugs by cutting a 16 pin wire wrap dip socket in two and stuffing the cable ends into the appropriate pin holes. Then just plug tha cables into the keyboard. This allows you to work on the keyboard or computer easily by simply disconnecting the keyboard.

30. KC9EZ writes I know there is a lot or interest in working cw with a computer. I, myself, use the program and interface of NU4V and it works very well. Now, if you want to ger rid of the extra E's, I's, and T's that go into your copy, try the QRM eliminator circuit found in 73 Magazine of April, 1983. Mine even knocked out the woodpecker. The circuit works as in the magazine with two corrections. C1 should be on the other side of the two diodes. and the junction of C2 and C3 should be grounded. Also for the ZX81, the matching transformer T1 is not used.

31. WB3JDP writes that he used the auto battery method to clear undesired traces from surplus keyboards (see BB number ??) without sucess. He resorted to drilling a series of holes across where the traces should have been. The result was lots of holes (and isolated keys).

32. Paul Hunter writes that his memory unit can be used to replace the 8K ROM but one must note that the character table is addressed by the SLC not by the CPU. If one wishes to modify the character table then it is just as well to place the unit inside the ZX81 case and attach it via a ribbon cable to the ROM socket. This way the SLC can access the character table (at 7.5 to 8K). If one wishes to change the ROM but leave the character table alone then the unit can be plugged on the back as usual. He will supply details to anyone interested if they write to the address given in the ad in this issue.

33. K8DD says that he has found that the transformer soldering irons and the low voltage irons tend to be much gentler with IC's than the old cord irons. However, he says that he is more concerned about bleeding the static charges off of himself that the irons.

BIBLIOGRAPHY

Alex Burr

The following is a list of selected articles in the larger ham magazines which discuss some aspect of RTTY. While few of them mention the Timex Sinclair computers directly, they do give a good introduction to the equipment and operating practices used by hams for RTTY.

HAM RADIO MAGAZINE

Active bandpass filter for RTTY	p. 46	April 79
An accurate and practical AFSK generator	p. 56	Aug 80
A vote for Baudot	p. 8	Mar 82
CW and RTTY digital audio filter	p. 60	Aug 83
Duplex Audio freq. gen. with AFSK	p. 66	Sept 79
LED tuning indicator for RTTY	p. 50	Mar 80
Phase coherent RTTY modulator	p. 26	Feb 79
A free RTTY tuning indicator	p. 74	Oct 81
RTTY zero beat indicator	p. 78	Oct 83
Slow ASCII	p. 6	June 81
FSK adaptor for SSB transmitters	p. 12	July 81
The XK2C AFSK generator	p. 58	Nov 80

73 MAGAZINE

RTTY Tuning The New Solution	p. 10	Mar 83
Unlock The New Electronic Mailboxes	p. 44	Apr 83
The New Communications: VHF Mailboxes	p. 48	May 83
Join The Packet Radio Revolution	p. 19	Sept 83
Be A RTTY Rembrandt	p. 28	Sept 83
Join The Packet Radio Revolution Part II	p. 20	Oct 83

QST

Graphics On RTTY	p. 11	Sept 83
Low Cost Modular Approch To RTTY	p. 16	Aug 83
Serial ASCII/Baudot Character Generator...	p. 20	July 83

CQ

See reveiw article elsewhere in issue.

THE ZX COMPILER

generates a machine code program from one written in an integer subset of Timex/Sinclair Basic

greatly increases program speed

machine code may reside anywhere in memory

commands/functions

DIM, FOR/NEXT, LET, IF/THEN, GOTO, GOSUB/RETURN, PRINT, LPRINT, PLOT/UNPLOT, POKE, STOP, FAST, SLOW, PAUSE, INPUT, RAND, CLS, COPY, SCROLL, AND, OR, NOT, ABS, SGN, USR, RND, PEEK, INKEYS, MOD, AT & TAB

- 256 variables (two letters)
- 26 single DIM arrays
- 16 bit arithemetic
- PRINT & INPUT in hex or decimal
- modulus function
- Hi and lo memory versions (23-32k & 12-16k)

both versions on one cassette **$25.00**

THE ZX ASSEMBLER & DISASSEMBLER

newly revised, one 4k program does all this:
- assembles standard mnemonics to machine code, source code in REM statements
- generates hex or decimal disassemblies to screen or printer
- allows hex or decimal editing of memory
- binary SAVE and LOAD files
- generates a REM statement of any size
- supplied in two versions, as compiler

both versions on one cassette **$20.00**

SOFTWARE ON EPROM

one 2764 (half blown) or one 2732A (please specify) fits cartridge below

ZX COMPILER	$30.00
ZX ASSEMBLER/DISASSEMBLER	$25.00

EPROM CARTRIDGE KIT

plug-in, uncased board holds one 2716, 2732, or 2764 mapped 0-16k, jumper selectable, kit includes board decoder chip, 28 pin socket, ZX connector, tailpiece, small parts, and instructions

Kit $18.00

THE ZX SERIAL PRINTER INTERFACE

plans and listing

RS-232 ASC II output thru MIC port. hardware cost about $10—256 bytes of machine code prints, characters, strings & listings. One connection within case. **$5.00**

CINAGRO SOFTWARE

(formerly Bob Berch)

155 Seventh St.

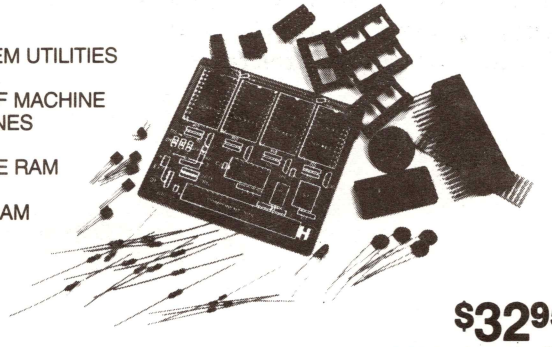
Rochester, N.Y. 14609

all prices include US shipping • NYS residents add 7% tax

Extend your ZX81/TS1000/TS1500: Add Memory that won't Forget!

➔ *DESCRIBED IN* **Radio-Electronics** *JULY/AUGUST 1983*

- ADD YOUR OWN SYSTEM UTILITIES
- BUILD UP A LIBRARY OF MACHINE LANGUAGE SUBROUTINES
- UP TO 8K NONVOLATILE RAM
- USE HM6116LP CMOS RAM OR 2716/2732 EPROM
- COMPATIBLE WITH 16K RAM PACKS



$32.95

plus $1.95 shipping and handling

► *READ THE REVIEWS:*

What a super product!...conceived and executed very nicely...and with quality components.
(SYNTAX QUARTERLY Winter 82)

8K Nonvolatile memory is a gem! It has so many possible uses...I recommend this board most heartily.
(OKLAHOMA S.U.G. Newsletter 1/3)

We found the documentation to be far superior to that (of) most hardware we've received.
(S.U.N. Newsletter Nov/Dec 82)

For versatility this is even better than an EPROM...ranks quite high on the list of "must-haves"...
(SYNC Magazine Mar/Apr 83)

Provides the user with instant software...an extremely versatile memory extension...
(Z-WEST June 83)

INTRODUCTION

This memory board is designed to fill the transparent 8K block of memory (from 8K to 16K) in a ZX81-16K system. This area of memory is an ideal place to store, either permanently or temporarily, machine language routines or data which are to be used by the BASIC system.

Sample utilities are included with the kit.

The use of HM6116LP 2K CMOS RAM memory IC's with their own reserve power supply means that routines stored in the RAM are nonvolatile — the RAM retains its memory even when the ZX81 is switched off or reset. Moreover, being RAM, the routines you store in the memory are easily modified. The lithium cell supplied with the board will maintain sufficient reserve power for almost ten years.

ASSEMBLY

Complete step-by-step instructions in a 20 page manual make assembly of the board easy. The kit (pictured above) is complete with a silkscreened solder-masked printed circuit board, all capacitors, resistors, transistors, sockets, connectors, integrated circuits, and the lithium cell. The board is supplied with one 2K CMOS 6116LP-3 RAM — it will accomodate three more for a total of 8K.

Complete kit with one 2K 6116LP-3	**$32.95**
Additional three 6116LP-3	**$18.00**
Bare pc board & manual	**$13.05**
Kit for EPROM use only	**$22.95**
Assembled & tested with 2K	**$47.95**
Assembled & tested with 8K	**$65.95**
Shipping & handling per order	**$ 1.95**

Send check or money order to the address below:

HUNTER, 1630 FOREST HILLS DRIVE, OKEMOS, MI 48864

73, W4GHV and K5XY

QZX presents the best software and hardware projects available for the radio amateur. While the contents are believed to be accurate, no warranty is extended on any program or project. No investigation has been made of the claims or reputation of any advertiser. It would be greatly appreciated if all correspondance requiring an answer would be accompanied by a self stamped envelope (SASE). Copyright by QZX, 1983. All rights reserved, Photocopying by permission only.

```
************************************
* Please send me QZX for 1 year,   *
*starting immediately. Enclosed is *
* $12.00 for first class           *
* mailing. (Those with addresses   *
* not in Canada, the U.S, or Mexico*
* send $19.00 in U.S. funds for    *
* air mail delivery.. Thanks(11/83)*
*                                  *
*  Name ..........................  *
*                                  *
*  Call ..........................  *
*                                  *
*  Address .......................  *
*                                  *
*  City ..........................  *
*                                  *
*  State.....  ZIP.....    *
************************************
```

QZX
2025 O'Donnell Dr.
Las Cruces, N.M. 88001

LAS CRUCES, NM PM DEC 23 1983 88001

Tom Arbster *****
Box 477
Bethany, Okla.
73008